Universidade de Aveiro
2016

Departamento de Eletrónica,
Telecomunicações e Informática

Jon Blancou Gaztañaga

eCall ++: chamadas de emergência rodoviárias melhoradas

eCall ++: Improved Emergency Call

Universidade de Aveiro
2016
Departamento de Eletrónica,
Telecomunicações e Informática

Jon Blancou Gaztañaga

eCall ++: chamadas de emergência rodoviárias melhoradas

eCall ++: Improved Emergency Call

" *Goroldiorik ez du dabilen harriak* "

— Basque proverb

Universidade de Aveiro
2016

Departamento de Eletrónica,
Telecomunicações e Informática

Jon Blancou Gaztañaga

eCall ++: chamadas de emergência rodoviárias melhoradas

eCall ++: Improved Emergency Call

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor Joaquim José de Castro Ferreira, Professor Adjunto da Escola Superior de Tecnologia e Gestão de Águeda da Universidade de Aveiro, e do Doutor José Alberto Gouveia Fonseca, Professor Associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dedico este trabalho à meus sobrinhos Asier, Nerea e Aratz.

o júri / the jury

presidente / president	Professor Doutor Paulo Bacelar Reis Pedreiras, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro
vogais / examiners committee	Professor Doutor Pedro Joaquim Amaro Sebastiao, Professor Auxiliar, ISCTE - Instituto Universitário de Lisboa
	Professor Doutor Joaquim José de Castro Ferreira, Professor Adjunto da Escola Superior de Tecnologia e Gestão de Águeda da Universidade de Aveiro

agradecimentos / acknowledgements

I would like to thank the University of Aveiro and the University of Deusto for given me this opportunity to study and to learn in another country and learning more about the tasks I like.
I also want to thank the professor Joaquim Ferreira and the researcher team for the help that they have given to me in this project.
Finally, I want like to thank all my family, old friends and new friends that I have made during my time in Aveiro for all the support that they have given to me.

Palavras Chave Sistemas de Transporte Inteligentes, Comunicações Veículares, Deteção automática de pasageiros, Video Streaming, eCall.

Resumo

Graças ao rápido desenvolvimento tecnológico e à grande adoção de aparelhos eletrónicos pessoais, como os smartphones, os monitorizadores de atividade física, etc., com uma diversidade de sensores e protocolos de comunicação, algumas aplicações no campo dos Sistemas de Transporte Inteligente (STI) podem beneficiar destes aparelhos. Por exemplo, os smartphones podem ser usados para navegação, como interface do condutor para aplicações veiculares colaborativas ou até mesmo para detetar acidentes. A deteção automática de acidentes é uma boa forma de melhorar os serviços de emergência rodoviários, contribuindo para uma diminuição do número de mortes.
Posto isto, a União Europeia (UE) lançou a iniciativa eCall, um sistema de deteção e notificação automático de acidentes que vai ser incorporado nos novos carros a partir de 2018. Os veículos antigos também poderão ser equipados com dispositivos eCall, mas provavelmente irão necessitar de equipamento adicional. Uma solução óbvia, para estes veículos, é usar os smartphones para detetar acidentes e disseminar chamadas de emergência.
Uma aplicação para smartphones com deteção de acidentes e uma implementação de um sistema eCall foi anteriormente desenvolvido no Instituto de Telecomunicações no contexto do projeto HEADWAY. No entanto, esta aplicação, bem como o sistema eCall original, não incluem alguns detalhes importantes como o número de passageiros existentes no veículo e um vídeo do local do acidente.
Este era o contexto subjacente à proposta presente nesta dissertação. O eCall++ é uma aplicação móvel que implementa o eCall nativo (deteção de acidentes e chamadas de emergência baseado em redes de telemóvel), bem como um sistema de alerta para os veículos na vizinhança do local do acidente, este último baseado no standard ETSI ITS-G5. Adicionalmente, a aplicação eCall++ quando detecta um acidente, transmite ao serviço de emergência o número de passageiros presentes no carro. Com a mesma aplicação, um carro próximo do sítio do acidente mas que não esteja envolvido no mesmo, pode ser contatado para transmitir um vídeo do local do acidente, através da rede veicular (comunicações ETSI ITS-G5). Este sistema irá promover uma melhor assistência aos acidentes rodoviários e facultar informação relevante sobre o tráfego às autoridades. Devido a razões de privacidade foi decidido não utilizar as câmaras dos veículos envolvidos no acidente.

Keywords

Intelligent Transport Systems,Vehicular Communication System, Passengger Detection, Video Streaming, eCall.

Abstract

Thanks to the fast technological development and the pervasive adoption of personal electronic devices, e.g. smartphones, activity trackers, etc., with a multitude of sensors and communication protocols, some applications in the field of the Intelligent Transport Systems (ITS) can take advantage of such devices. For example, smartphones can be used for navigation, as driver interface for cooperative vehicular applications or even to detect accidents. Automated accident detection is a potential way to improve the road emergency services, thus decreasing the number of fatalities.

For this purpose, the European Union (EU) has launched the eCall initiative, an automated accident detection and notification system, that will be mandatory for new cars in 2018. Legacy vehicles can also be retrofitted with eCall devices, that may require some wiring to install. An obvious solution, for legacy vehicles, is using the smartphone to detect accidents and disseminate emergency calls.

A smartphone based accident detection algorithm and eCall application was already developed at Instituto de Telecomunicações in the scope of the HEADWAY project. However, this application, as the native eCall, missed to offer some important features, notably information on the number of passenger in the vehicle and a video stream of the accident site.

This was the context behind the proposal presented in the dissertation. The eCall++ is a smartphone application that implements native eCall (accident detection and cellular network based emergency call), plus warning dissemination to vehicles in the vicinity of the accident site, based on ITS G5 communications. Additionally, the eCall++ application running on a vehicle that is involded in an accident also transmits to the emergency service the number of passengers in the car. An eCall++ application running on a car close to the accident site, but not involved in the accident, can be asked by the emergency service to transmit a video stream of the accident, using the vehicular network (ETSI ITS G5 communications). This will promote a better assistance to road vehicle crash location and to facilitate relevant information about traffic incidents to authorities. Due to privacy reasons it was decided not to use the camera of the vehicles involved in an accident.

Palabras clave

Sistemas Inteligentes de Transporte, Detección de pasajeross, Video Streaming, Sistemas Comunicaciones Vehiculares, eCall.

Resumen ejecutivo

Gracias a la gran velocidad de desarrollo de las tecnologías y la adopción generalizada de los aparatos electrónicos, tales como smartphones, seguidores de actividad, etc., que son equipados por multitud de sensores y protocolos de comunicación, algunas aplicaciones para el ámbito de los Sistemas Inteligentes de Transporte (ITS) pueden sacar partido de este tipo de dispositivos. Los smartphones pueden ser utilizados como sistemas de navegación, sistemas de cooperación en ámbito vehicular o simplemente detección de accidentes. La detección de accidentes automatizada es una gran solución para mejorar los sistemas de emergencia en las carreteras y descender el número de victimas en ellas.

Con este fin, la Unión Europea (UE) está promoviendo la iniciativa eCall. Se trata de un sistema autónomo de detección y notificación de accidentes que será obligatorio para los nuevos automóviles a partir del 2018. Sin embargo, los modelos mas viejos podrán usar el sistema eCall instalando aparatos específicos. Una de las soluciones mas simples para incorporar el eCall es el smartphone.

Una aplicación para smartphones provista de una algoritmo de detección de accidentes y sistema eCall fue desarrollada en el Instituto de Telecomunicações para el proyecto HEADWAY. Sin embargo, esta aplicación, al igual que la aplicación nativa del eCall, no esta dotada de importantes funcionalidades, tales como video streaming del lugar accidentado o la detección de pasajeros.

Este es el contexto previo al proyecto propuesto en esta tesina. El eCall++ es una aplicación que implementa el sistema nativo de eCall (detección de accidentes y llamadas de emergencia en redes móviles), añadiendo avisos del estado de la carretera a los vehículos próximos a un incidente basado en comunicaciones ETSI ITS G5. Además, la aplicación eCall++ envía el número de pasajeros en el vehículo en caso de accidente. Por otro lado, si un vehículo no involucrado en el accidente esta en las proximidades a este puede recibir una solicitud de video streaming y de esta manera informar de la gravedad del incidente. Como se puede deducir, esta aplicación proveerá información necesaria para localizar accidentes y facilitar información importante acerca del estado del tráfico. Debido a asuntos de privacidad se ha decidido no usar la cámara de los vehículos involucrados en los accidentes.

Hitz gakoakadmin

Android, bidaiari detekzioa, bideo streaming-a, Garraiorako Zentzuko Sistemak, eCall.

Laburpena

Azkenaldian garatu diren teknologiei ezker eta gureganatu diren gailuei ezker, smartphoneak edota aktibitate jarratzailei ezker adibidez Garraiorako Adimentsuentzako Sistemek (GAS) aplikazioei probetzu atera ahal diete. Smartphone-ak nabigazio sistema bezala, eremu ibilgailuetan sistema kooperatibo bezala edo istripu detekzioentzako erabili daitezke. Istripu-detekzio mekanismo automatizatua erantzun ona da larrialdi sistemak obetzeko eta trafiko istripuetako biktimak jaisteko.
Helburu honekin, Europar Batasunak (EB) eCall ekimena bultzatzean dabil. Ekimen hau istripu detekzioan eta jakinarazpenean datza eta 2018tik aurrera EUko auto berrietan imposatzen den sistema izanen da. Beste aldetik, auto zaharrek gailu espezifikaok instalatu ahalko dituzte eCall-a erabiltzeko. Soluzio sinple bat smartphone-ak helburu honekin erabiltzea da.
HEADWAY proiekturako Instituto de Telecomunicações-en istripu-detekzio algoritmodun eta eCall sisteman hornitutako aplikazioa garatua izan zen. Hala ere, aplikazio honek, eCall natiboa bezala, ez ditu funtzionalitate ugari garatzen, bideo streaming-a edo bidaiari detekzioa adibidez.
Hau da tesina honen aurreko testuingurua. eCall++-a eCall sistema natiboa, istripu detekzioa eta mobil sareaetatik egindako larrialdiko deiak, garatzen duen aplikazioa da. Aplikazio honek, errepideko informazioa eguneratzen dio gidaiariari. Istripua izatekotan larrialdi sistemei bidaiari kantitatea eskuratzen dio ETSI IT G5 komunikazioei ezker. Beste alde batetik, ibilgailu bat istripu batetik gertu baldin badago bideo streaming-erako eskaera izan dezake. Era honetan aplikazio honek istripua aurkitzeko eta trafikoari buruzko informazioa eskuratzeko aukera ematen du. Pribatutasuna dela eta proiektu honetan istripuan nahasita dauden autoen kamarak ez dira erabil.

Contents

Contents . . . i

List of Figures . . . iii

List of Tables . . . v

Glossary . . . vii

1 Introduction . . . 1
1.1 Motivation and context . . . 1
1.2 Goals . . . 2
1.3 Structure of the document . . . 3

2 Related work . . . 5
2.1 eCall . . . 5
2.1.1 eCall chain . . . 6
2.2 Intelligent Transport Systems . . . 8
2.2.1 Main benefits of Intelligent Transport Systems . . . 9
2.3 Vehicular Communication Network . . . 9
2.3.1 Cooperative Awareness Message (CAM) and Decentralized Environmental Notification Message (DENM) messages . . . 12
2.4 Smartphones . . . 15
2.4.1 Android . . . 15
2.5 Related projects and applications . . . 16
2.5.1 Applications available in platform repositories . . . 16
2.5.2 Available passenger detection systems . . . 19
2.6 Conclusion of related work . . . 21

3 eCall++ architecture . . . 23
3.1 eCall ++ architecture . . . 23
3.2 Requirements . . . 26
3.3 Use cases of eCall++ . . . 27
3.4 IT^{2}S platform as an On Board Unit (OBU) and Road Side Unit (RSU) . . . 30
3.4.1 IT^{2}S platform architecture . . . 30
3.5 Smartphone as an application Unit . . . 31
3.6 Main guidelines of the application . . . 33
3.6.1 Graphical User Interface . . . 33

3.7 Proposed implementation 34
3.7.1 Graphical User Interface (GUI) mockup 35

4 Implementation of the eCall++ 41
4.1 Communication interface 41
4.2 eCall++ implementation 43
4.2.1 Architecture 44
4.2.2 Design 56

5 Test and validation 63
5.1 Passenger detection mechanism test 63
5.1.1 Passenger detection results 64
5.2 Video streaming mechanism 69
5.2.1 Video streaming quality test 70

6 Conclusion 73
6.1 Limitation and challenges of eCall++ for vehicular networks 73
6.2 Future work 74

Bibliography 77

List of Figures

2.1 eCall chain. *Adapted from: [6]* 6
2.2 eCall service chain domains [8] 8
2.3 C2C-CC vehicular network reference architecture. *Source:* [12] 10
2.4 IEEE and ETSI vehicular communication protocol stack. *Source:* [13] 11

3.1 eCall++'s scenario 24
3.2 eCall ++'s use cases diagram 28
3.3 ITS platform architecture. *Adapted from:* [47] 31
3.4 Overall system integration target implementation 32
3.5 Received layout mockup 36
3.6 Report layout mockup 36
3.7 Manual Road Hazard Warning (RHW) block diagram. 37
3.8 OBD-II layout mockup 37
3.9 Settings layout mockup 38
3.10 Passenger detection layout mockup 38
3.11 Passenger detection block diagram 39
3.12 Sending video streaming block diagram 39

4.1 OBD-II data frame 42
4.2 RHW data frame from the IT^{2}S platform to the smartphone 42
4.3 RHW data frame from the smartphone to the IT^{2}S platform 43
4.4 eCall++'s Activity diagram 45
4.5 User Interface (UI) navigation sub Activity state machine. *Adapted from:* [48] . 47
4.6 Display RHW action sub Activity diagram. *Adapted from:* [48] 49
4.7 Passenger detection Activity diagram 51
4.8 Not continuous live video streaming diagram 53
4.9 Class diagram 55
4.10 Receive Activity 56
4.11 Settings Activity 57
4.12 Report Activity 58
4.13 OBD-II Activity 58
4.14 Passenger detection Activity 59
4.15 Count Down Activity 59
4.16 Global Positioning System (GPS) enable dialogue box 60
4.17 Good bye Activity 60
4.18 OBD No connect Activity 61

5.1 Scenario of passenger detection test 64
5.2 Out of focus photograph for passenger detection 66
5.3 Photograph where the back seat passengers are covered up by the front seats . . 67
5.4 Not good face angle photograph 67
5.5 Error detection application 68
5.6 Out of focus photograph for passenger detection 70

List of Tables

2.1 Multichannel Vehicular ad-hoc Network (VANET) standards: United States vs. Europe. *Adapted from:* [13] 12
2.2 CAM Use cases. *Source:* [14] 13
2.3 DENM use cases, triggering and termination conditions. *Source:* [14] 14
2.4 Mobile application comparison 19

4.1 RHW use cases in eCall++. *Source:* [15] 43

5.1 Passenger detection test during the day light 65
5.2 Passenger detection test during night light 66
5.3 Test datas and arithmetic average comparison 69
5.4 Results of non-IP video streaming test 71

Glossary

AAD Autonomous Accident Detection

ADA Accident Detection Algorithm

ADC Analog to Digital Converter

AOSP Android Open Source Project

API Application Programming Interface

AU Application Unit

BSM Basic Safety Message

BSA Basis Set of Applications

C2C-CC Car to Car Communication Consortium

CAM Cooperative Awareness Message

CCH Control CHannel

CEN European Committee for Standardisation

CLI Called Line Identificartion

DB Data Base

DAC Digital Analog converter

DENM Decentralized Environmental Notification Message

DSRC Dedicated Short Range Communications

EG eCall Generator

EMS Emergency Medical Systems

ETSI European Telecommunications Standards Institute

EU European Union

EC European Commission

FIR Far-Infrared Ray

FPGA Field-Programmable Gate Array

GLONASS Global Navigation Satellite System

GPS Global Positioning System

GUI Graphical User Interface

GSM Global System for Mobile communications

HEADWAY Highway Environment Advanced Warning System

HMI Human to Machine Interface

HOT High Occupancy Toll

I2V Infrastructure to Vehicle

ICSI Intelligent Cooperative Sensing for Improved Traffic Efficency

IDE Integrated Development Environment

IEEE Institute of Electrical and Electronics Engineers

IDC International Data Corporation

ITS Intelligent Transport Systems

LTE Long Term Evolution

MAC Media Access Control

MIB Management Information Base

MNO Mobile Network Providers

MLME MAC Layer Management Entity

MSD Minimun Set of Data

NFC Near Field Communications

OBU On Board Unit

ODS Occupancy Detection System

OFDM Orthogonal Frequency-Division Multiplexing

OHA Open Handset Alliance

OS Operating System

PER	Packet Error Rate
PDU	Protocol Data Unit
PHY	Physical
PID	Parameter Identificator
PODS	Passive Occupant Detection System
PSAP	Public Safety Answering Point
RF	Radio Frequency
RHW	Road Hazard Warning
RSU	Road Side Unit
SAE	Society of Automotive Engineers
SDK	Software Development Kit
SCH	Service CHannel
SIFT	Scale Invariant Feature Transform
SMS	Short Message Service
UI	User Interface
UML	Unifield Modeling Language
UMTS	Universal Mobile Telecommunications System
USB	Universal Serial Bus
UTC	Coordinated Universal Time
UX	User Experience
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	Vehicle to Anything
VANET	Vehicular ad-hoc Network
VC	Vehicular Communication
VCN	Vehicular Communication Network
VIN	Vehicular Identification Number
WAVE	Wireless Access in Vehicular Environments
WSMP	WAVE Short Message Protocol
RMA	Resource Manager Application

CHAPTER 1

Introduction

1.1 Motivation and Context

Since the beginning of time, the humans have been trying to transport goods and persons in a comfortable and safe way trough the world. Different ideas emerged and evolved with the contribution of many people to reach this goal. But it was not since Henry Ford invented the assembly line that the automotive vehicle became an essential machine for the daily life.

The vehicles have been changing their aerodynamic form and inside appearance for increasing their efficiency, security, comfort or just fashion. The vehicles have been equipped also with new technology because of the same reasons. The technology in the vehicles has changed a lot in the last decades. The automotive industry keeps an eye on daily application technology and they are introducing the smartphones and tablets in the vehicular environment.

These devices are the aggregation of different technologies in a single one. Today smartphones are equipped with sensors (accelerometer, gyroscope, magnetometer, etc), big screens, dual-core processors, location systems such as GPS and Global Navigation Satellite System (GLONASS), powerful cameras and different communication systems like Global System for Mobile communications (GSM), Universal Mobile Telecommunications System (UMTS), Long Term Evolution (LTE)), Bluetooth, Near Field Communications (NFC) and Wi-Fi. The process of the introduction of smartphone devices in our everyday life is because of the simple User Experience (UX) and because they are provided by a powerful hardware integrated in a small and portable device.

Due to its powerful hardware and software smartphones are used for several purposes. The largest part of these purposes are for daily uses but also the transportation systems companies are using theses type of technologies to make more efficient transportation of goods.

During the last years the European Union (EU) also has proposed to develop this type of technology to increase safety and to get more efficient traffic at the roads. At 2014 [1] almost 25700 road fatalities were reported in the EU and the European Commission (EC) want to decrease this number of accidents. One of those innovative initiative of the EC is the eCall that will decrease the arrival time of medical aid when an accident occurs on a road.

To tackle this issue, the Vehicle Communication Systems have been also introduced into vehicles. With the Vehicular Communication Systems, development the Intelligent Transport Systems (ITS) concept is emerging. The aim of this system is to improve safety, reduce congestion and save fuel. The

Vehicular Communication (VC)s are booming as a research topic. Numerous universities, institutes, auto-makers and telecommunication companies are researching and developing solutions to be deployed for massive scale. The European Telecommunications Standards Institute (ETSI) [2] and the Institute of Electrical and Electronics Engineers (IEEE) [3] have already published standards to be followed in this area.

It is assumed that the more information is sent between vehicles, the better decisions the drivers will make. With more information drivers could decide the best route to take, or even carefully approach a certain location within their route, knowing that the location is marked as unsafe.

Today cars offer some ITS services like on-board computers with GPS navigation systems, weather and entertainment applications or Autonomous Accident Detection (AAD) system. In older and lower quality vehicles however smartphones are being used to provide those services to the driver. Smartphones have become a valuable solution to boost ITS in automobiles.

This project aims to offer services to VC. For this goal, this dissertation develops an eCall system application that takes advantage of the device's hardware resources, to provide the application of a AAD mechanism along with a help of request mechanism, passenger detection and video streaming mechanisms.

1.2 GOALS

The main goal of this dissertation is to develop an eCall system composed by Accident Detection Algorithm (ADA) mechanism, video streaming and passenger detection mechanisms, based on a smartphone, to detect road accidents and perform a help request based on the eCall requirements. The eCall++ is integrated in a smartphone to demonstrate the potential of vehicular communication networks in case of an accident occurs in a road. This dissertation follows the ETSI and the IEEE standards for vehicular communications.

The dissertation target, is to develop more functionalities than a normal eCall application. The eCall++ must be able to detect the passengers, send video from the smartphone to the vehicular communication system, and to exchange road safety and efficiency related messages with other vehicles. eCall++ should provide the following features:

- **GUI:** the GUI must be designed considering the contest of use of the application, the driving act. It should also provide to the user the means to interact with the system.
- **RHW report:** this project should incorporate to the application the functionality of sending a location, time based messages notifying road related information. RHWs are exchanged messages in the vehicular systems implemented by ETSI. These messages allow the user to send the information about a traffic event, such as traffic jams, road hazards, accidents, broken down vehicles or reduced visibility.
- **eCall system:** as soon as the mobile application detects a car accident, the eCall system should send a rescue message. The accident should be detected through the smartphone device hardware. The user should be able to abort the rescue message during a countdown.
- **Passenger detection mechanism:** the eCall++ must be able to detect the number of passengers inside the vehicle without requiring action from the user or the occupants. With this

mechanism, the smartphone is able to inform to the emergency services how many occupants are inside the car. The smartphone hardware resources should be used for this function.

- **Video Streaming mechanism:** this service must use the smartphone as an on board camera that records and streams the view through a vehicle's windscreen. This function may provide video evidence in the event of an accident on road situation, and transmit video streams to the emergency services and road operators. The transmission is requested by authorities to the car in the vicinity of an accident to better coordinate the emergency response.

1.3 STRUCTURE OF THE DOCUMENT

This document is written into six chapters. In the first one (1), the contextual introduction is presented with the intention to make understood why the ITSs are so important nowadays and the main goals of this dissertation.

The second chapter (2), presents the eCall chain where the eCall++ is based. The ITS and Vehicular Communication Networks are described due to implication in the eCall chain. Following, Android Operating System (OS) is presented to introduce the eCall++ environment. This chapter finishes with a review of the related work that supports this dissertation. In this research several application are described and compared.

The third chapter (3) presents the proposal of the eCall++. In this proposal the architecture of the eCall++, drafts and guidelines of a solution to be implemented and tested are defined.

In the fourth chapter (4) all the relevant aspects of the application implementation is described. The decisions made in this chapter are based on the previous chapters.

The fifth chapter (5) advances the tests and validation of the eCall++. These tests are the passenger detection mechanism and video streaming quality tests.

The sixth chapter (6) presents the dissertation conclusion, possible future improvements to eCall++ and the limitations of the eCall++ on a vehicular network environment.

CHAPTER 2

Related work

In the last decades the development of VANETs, vehicular network technology and driving assistance technology have attracted a great investment from academics, industry, and governments worldwide in order to leverage the benefits of vehicular communications to road safety, and improved traffic flow. The EC also had its eye on these types of technologies and has promoted the eCall. This is the starting point of the work described in this dissertation, exposed in section 2.1, and therefore the state of art of this dissertation starts describing this concept.

In the second section, the ITS systems concept and the Vehicular Communication Network (VCN) terms are depicted due to the close relationship to the eCall and, consequently to the eCall++.

Afterwords, the smartphone environment, focusing on Android, is presented with the purpose to understand the market penetration and technical characteristics of mobile platforms.

Finally, as a result of a search, the related projects and mobile applications are compared for the goals to discover the market situation and the characteristics of projects in this category. A search is also performed in the passenger detection field to assess what could be implemented in the smartphone for that purpose.

2.1 eCall

Thousands of fatalities happen on the roads every year in EU and the EC has decided to promote the eCall to decrease the number of deaths on the roads. Some of this misfortunes could be decreased if the reaction time of the Emergency Medical Systems (EMS) is improved. A considerable time is lost when an accident occurs and an injured person receives the proper medical aid. This lost time is directly related to the probability of death and trauma. This time could be decreased if help is requested immediately after the accident event happens.

A quicker and better response from the assistance is possible if the EMSs has the appropriate information as the eCall project proposes.eCall is an automatic accident detector that in the presence of an accident automatically requests help to the EMS through the European 112 emergency number. This information is presented in this section. The EC has calculated [4] that 2,500 lives per year are expected to be saved throughout Europe with eCall.

The EC, the 18th of April of 2015 [5], postponed the deployment of eCall in new cars after 2018 in EU.

Following the eCall chain is described with more details to understand the bases of this dissertation.

2.1.1 eCall chain

On the eCall chain several technological aspects must be combined with all the intervening parts to perform this system. The chain of eCall is described in the figure 2.1.

The eCall chain is composed by three parts, the car manufacturers, the Mobile Network Providers (MNO) and the participant countries from the EU. The EC has decided regulations to mandate the deployment of technology in eCall and updating of the three parts. Each participant has to take care of the development of the involved technology and also that these technologies are in accordance with the eCall specifications.

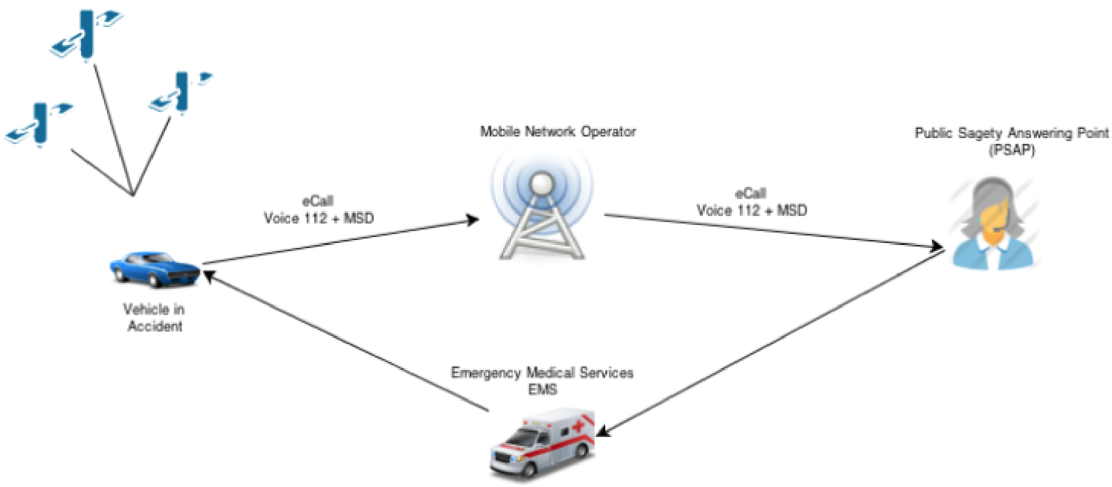

Figure 2.1: eCall chain. *Adapted from: [6]*

In the figure 2.1 is described how the eCall works when an accident occurs. The eCall transmits two type of messages, a voice call and an Minimun Set of Data (MSD), to the most suitable Public Safety Answering Point (PSAP) through the MNO. The solution adopted for the MSD transmission, is an in-band modem that transmits data in the voice channel.

The MNO, in order to support this type of connection, is obliged to upgrade their networks before 2018 [5]. EC also mandates the upgrade of the PSAPs from the member countries for a faster MSD information analyse. The EMS could also be dispatched faster.

If the request is sent after the accident happens a quicker reaction is possible. The EMS could give an efficient medical aid if the information defined in the MSD is useful.

The information from the MSD [7] can be divided in two categories: mandatory and optional. The mandatory information should contain information to help to speed up the EMS arrival to the accident scene. The optional information is not necessary to send in the MSD, but it provides to the EMS more information about the car environment in an accident. The eCall has been standardised by the European Committee for Standardisation (CEN) EN 15722. The MSD [7] should be sent in a 140 Bytes containing the following information:

- **Message identifier (Mandatory):** MSD format version (later versions to be backwards compatible with existing versions).
- **Activation (Mandatory):** whether the eCall has been manually or automatically generated.
- **Call type (Mandatory):** whether the eCall is real emergency or test call.
- **Vehicle type (Mandatory):** passenger Vehicle, buses and coaches, light commercial vehicles, heavy duty vehicles, motorcycles
- **Vehicular Identification Number (VIN) (Mandatory)**.
- **Vehicle propulsion storage type (Mandatory):** this is important particularly relating to fire risk and electrical power source issues (e.g. Gasoline tank, Diesel tank, Compressed Natural Gas (CNG), etc.)
- **Time stamp (Mandatory):** Timestamp of incident event.
- **Vehicle location (Mandatory):** determined by the on-board system at the time of message generation. It is the last known vehicle's position (latitude and longitude).
- **Confidence in position (Mandatory):** this bit is to be set to "Low confidence in position" if the position is not within the limits of +/-150m with 95% confidence.
- **Direction (Mandatory):** helpful to determine the carriageway vehicle was using at the moment of the incident.
- **Recent vehicle location (Optional):** vehicle's position in (n-1) and (n-2).
- **Number of passengers (Optional):** number of fastened seatbelts.
- **Optional additional data (Optional):** in some cases, optional data may be available in the MSD, for example, the vehicle manufacturer discretion. This data incorporates a tag for the identification in the beginning of the optional data. This data will be registered and maintained. PSAP will have free access to the registered data.

As it is possible to read the MSD could be extended with optional set of data, not exceeding the available number of bytes.

The eCall must work in all the participant countries simultaneously. This means that any vehicle from any participant country must be able to work in the other participant countries, i.e, if a driver has an accident with his vehicle, in a foreign country, the eCall is designed to connect to the nearest PSAPs of that foreign country.

In the eCall Driver Group recommendations, the eCall service chain [8] can be found. As it is depicted in figure 2.2 there are six main domains of this chain. Following each domain is described:

- **Vehicle eCall Triggering System:** the trigger is composed by sensor that should detect front, side, rear and roll crashes. This sensor is composed by the airbag module and/or a combination of other sensor data, such as, gyroscope, radar, or speed detector. The triggering threshold is based on speed variations or also be sent as optional help PSAPs predict the likeliness of serious injuries. As it is previously mentioned the eCall can also be activated in a manual way.
- **eCall Generator (EG):** in-vehicle software triggers the eCall, it provides the necessary information from the triggering system. Following the in-vehicle communication module initiates 112 call and sends the MSD through the in-band module.

- **EG to MNO:** the network receives the 112 call and the MSD.
- **MNO:** the mobile network operator (MNO) enriches the 112 call with Called Line Identificartion (CLI), MSD and cellular location.
- **MNO to PSAP:** forwards the enriched 112 call to the appropriate PSAP.
- **PSAP:** answers 112 voice call, decodes and visualises cell location and PSAP.

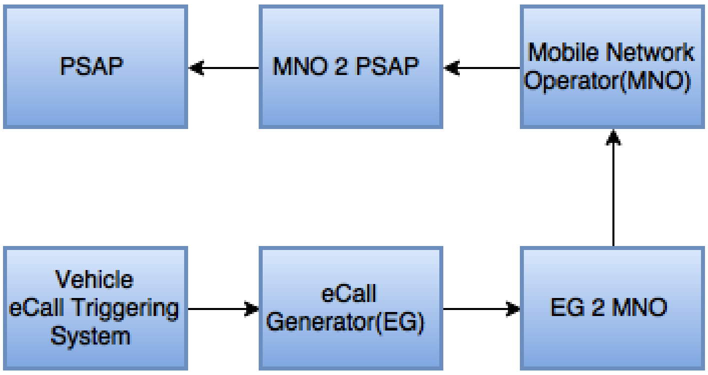

Figure 2.2: eCall service chain domains [8]

The eCall is based on the ITS and vehicular communication network concepts and it is hard to understand the eCall++ environment without having a view on these terms. The eCall chain uses ITS in an vehicular communication network architecture. These technologies are further described in the 2.2, and 2.3 sections to establish a state of the art of this dissertation.

2.2 INTELLIGENT TRANSPORT SYSTEMS

The Intelligent Transport System (ITS) was born in 1980 [9] from a small group of transportation professionals as a set of telecommunication and informatics solutions for the problems that could appear in the transport of the people and goods. Since the 80s the world of the road transport has had computing and communication revolution. Nowadays, it is totally involved in the Information Age. A glance to the history shows us that the evolution of transport systems had the principal intention of being faster, safer, and cheaper, becoming more efficient step-by-step. The internal combustion engine also, was introduced to the transport world to archive these goals.

The ITS are extensively integrated in daily life. Technologies such as electronic tolling, navigation product or emergency notification systems in commercial vehicles are totally normal in nowadays roads. Another social unrest where the ITS has a strong presence is the environmental pollution. Irrespective of the type of fuel to generate power on the vehicles, this type of internal combustion engine releases toxic gases to the atmosphere. These toxic gases contribute to impoverish the quality of the air, increasing the green house effect and inflicting respiratory problems and harming people's health.

As it is seen the ITS affects directly to the peoples quality of life. The ITS likewise has changed the several important aspects, connecting them more efficiently. The economy, time cost, health, environmental pollution has changed with these new systems [10].

2.2.1 MAIN BENEFITS OF INTELLIGENT TRANSPORT SYSTEMS

The main goals of the ITS are safety in the transportation, pollution reduction and time/cost efficiency in transport. In accordance with [11] the benefits of the ITS could be divided in the following way:

- **Safety increase:** one of the most important goal of the ITS is the security improvement for the vehicles and the transport infrastructure on the road. The evolution of technology permits the development of new systems to be integrated to the vehicles and highways. This helps the driver through navigation system, collision prevention system, driver fatigue detection system and error detection system. Making use of VC, it is possible to keep drivers informed and make them aware of the possible dangers in their route, allowing a better decision. The safety is a priority for the infrastructure managers. They use different type of ITS, to coordinate traffic controls, controls in the access way to the highway, information panels, traffic accident detection systems etc.
- **Environment preservation:** the use of technologies related to ITS, helps to provide real time information to the driver. This issue concerns to drive shorter routes, avoiding traffic jams, reducing the fuel consume and the level of ejected pollution.
- **Transport efficiency: (Mobility/Productivity)** with the use of the ITS technologies the information such as vehicle location, weather forecast, traffic condition and more are available. This type of information helps the drivers to know where is the transport and to change the route to select a faster way to arrive to the destination. All this type of information and techniques could change to a better and more efficient service. The transport efficiency is directly related to the environment preservation and the time/cost reduction.

The Electronic Toll Collection is a popular ITS application with the principal objective to reduce the delay of the toll queue at the highways. It allows a more efficient payment method and therefore a more traffic flow in motorways.

The ITS are also able to enable vehicles to be aware of road intervening elements. These systems are able to inform about the exact position, velocity and acceleration of the proximate vehicles, as well as to access to information about what to expect down on the route. These technologies allow the driver to do a better planning and decision making. This information contributes for safety and efficiency increase on the road.

The VCNs are really related to the ITS, due to these technologies are used in these type of networks. These networks, that are depicted in the next section 2.3, are able to transmit any type of digital information between vehicles using ITSs.

2.3 VEHICULAR COMMUNICATION NETWORK

VCNs are created to allow vehicles to warn about the intervening elements of the road in high-speed mobile environment. With short and medium wireless communication technology the highway driver and system could know the exact position, speed and acceleration of other vehicles. So directly, as the ITS do, the VCNs increase safety and efficiency accessing information about what to expect down on the route. Because of this they allow a better planning and decision-making in this type of networks.

Due to the high-speed mobile environment the VCNs are different to the current networks that usually are static. To complete this type of communication, the VANET emerged.

The VANET consists in moving vehicles with wireless communication interfaces. The VCN is a work in process and there are research projects under development that produce new ideas that could update the VANET concept. One of the main references for the VC is Car to Car Communication Consortium (C2C-CC) published by Z. Moustafa [12]. Figure 2.3 presents this type of architecture.

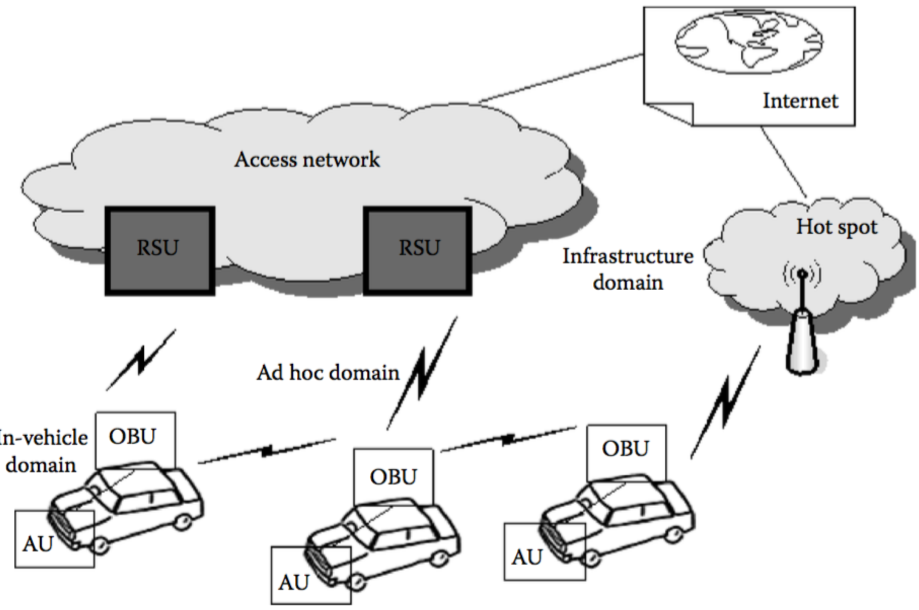

Figure 2.3: C2C-CC vehicular network reference architecture. *Source:* [12]

This architecture presented in the figure 2.3 distinguishes three different areas, the in-vehicle, the ad-hoc and infrastructure domain. The network in vehicle domain is composed by two units: the OBU and the Application Unit (AU).

On one hand, the OBU is a device with communication abilities. On the other hand, the AU is a device executing applications that take advantage of OBU's communication capabilities. These two devices could be in the same physical device. Moreover, the AU could be a single device, such as a smartphone, that can be dynamically attached or detached to/from the OBU. The communication between these two units could be wired or wireless.

Observing the figure 2.3, two units, OBU and RSU, compose the ad-hoc domain. This domain is defined where VANET concept is integrated. The RSUs are static units along the road and can be attached to an infrastructure network; witch could be connected to the Internet. The RSUs are able to connect with another RSU or with OBUs. The ad-hoc concept implies that the OBUs and the RSUs could behave as a network node. In this way, this concept allows the units to receive and redirect information through network.

The infrastructure of a C2C-CC vehicular network could be composed by RSU s and hot spots. OBU are able to communicate with Internet because of the communication with the RSU through hot spots, and the connection of RSU with Access Network. The hot spots could help in this communications using different technologies, for example, GSM, UMTS, WiMax and LTE, if these are integrated at the OBU.

The two communications directions that are in this network are the Vehicle to Vehicle (V2V) and Infrastructure to Vehicle (I2V) (or Vehicle to Infrastructure (V2I)). The communication that integrates both directions is called Vehicle to Anything (V2X).

The interest of industry and scientific community is the cause of the development in the area of VC. These new projects that are been developing are resulting in new standards that became in specification to be followed. There are two main protocols stack to be followed for V2X communications. On the one hand, the standard from ETSI (Europe), and on the other hand, the standard from IEEE (USA). Figure 2.4 represents the two protocols of these organizations.

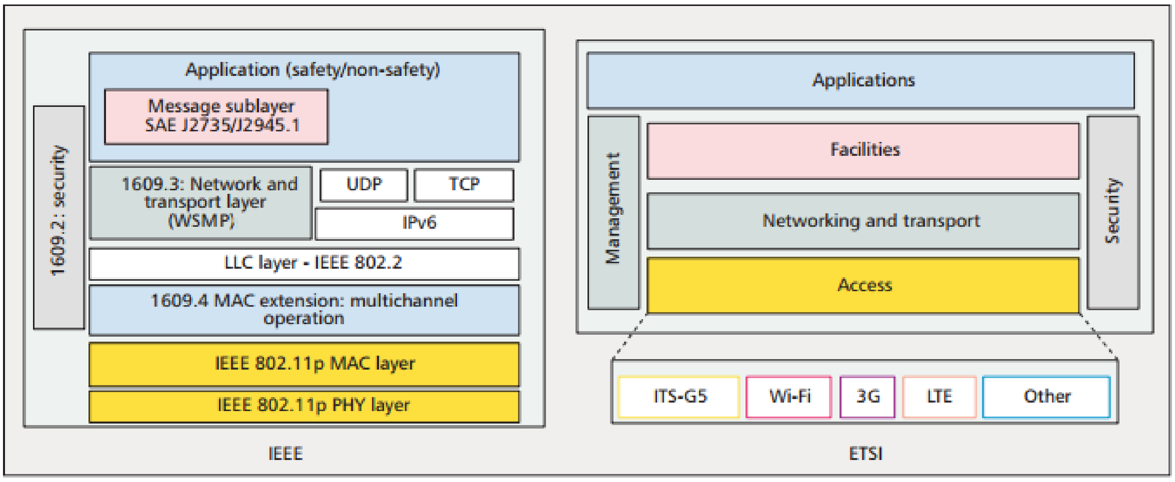

Figure 2.4: IEEE and ETSI vehicular communication protocol stack. *Source:* [13]

A figure 2.4 illustrates the European and American protocol stack architecture and the fact is that they share some design options. Even if these protocol stacks are not definitive yet, they could be changed and are still on going. The most important differences between them are described in the next table 2.1.

	European Union	United States of America
Frequency rate	5.85-5.905 GHz	5.85-5.925 GHz
Control channel frequency	5.9GHz	5.89GHz
Number of Service channels	4(CH172, CH174, CH176, CH 178)	6(CH172, CH174, CH176, CH 180, CH 182,CH184)
Main Standardization bodies	ETSI, ISO/CEN, CEPT	IEEE, SAE International, FCC
PHY/MAC Layers	Multi access technologies, among which ETSI ITS-G5, same as 802.11p	IEEE 802.11p, 16099.4
Higher Layers	ETSI ITS station: Facility Networking and Transport, Security layers	IEEE 1609.0 169.2 1609.3
Application / message formats	ETSI	IEEE 1609.x , SAE International
Service announcements	SAM (Service Announcement)	WSA (WAVE Service Advertisement)
Main Safety massages	CAM, DENM	Basic Safety Message (BSM)

Table 2.1: Multichannel VANET standards: United States vs. Europe. *Adapted from:* [13]

The main differences between two protocols are the radio spectrum allocated and the lead safety messages. The allocated spectrum of ETSI is 50MHz and the allocated spectrum of IEEE is 75MHz.

The most important differences between these two protocol stacks are the main safety messages that are used. In the ETSI stack, these type of messages are implemented in the Facility layer. On one hand, ETSI uses two type of main safety messages; the CAM and the DENM. Those messages structure and purpose are explained at section 2.3.1. The IEEE WAVE standards also include similar messages.

The ETSI stack also predicts the use of different wireless technologies, for example Wi-Fi, UMTS, LTE, ITS-G5 (ETSI vehicular communication standard), among others to establish communications. On the other hand, the IEEE uses the BSM that is implemented in the Message sublayer using the Society of Automotive Engineers (SAE) J2735/J2945.1 standards.

2.3.1 CAM and DENM messages

The VCN contributes to exchange information between different vehicles. This information is predefined by a standard with the finality to send fixed messages from the OBUs. The RSUs could participate in V2X networks receiving these messages. This project implements the ETSI Facilities layer, as it is described on section 4.2.1, where this type of messages are generated and managed.

Cooperative awareness message (CAM)

In the light of ETSI TS 102 637-2 [14], the CAM messages inform about the presence, the position and the basic status of near by ITS station (RSUs and OBUs). These stations are located in a single hop distance, ergo, the maximum allowed distance for the communication between two stations. These type of messages could be made by ITS devices in V2X network.

The following table 2.2 contains the Basis Set of Applications (BSA) use cases based on CAMs and the corresponding timing requirements:

Use case	**Min Frequenzy(Hz)**	**Min latency (ms)**
Emergency Vehicle Warning	10	100
Slow Vehicle Indication	2	10
Intersection Collision Warning	10	100
Motorcycle Approaching Indication	2	100
Collision Risk Warning	10	100
Speed Limits Notification	1 to 10	100
Traffic Lights Optimal Speed Advisory	2	100

Table 2.2: CAM Use cases. *Source:* [14]

The cases where the CAM messages are defined in the previous seven cases from table 2.2. CAM messages rate time of generate and transmission is 0,1 seconds, the minimum latency required in the other hand is 100 milliseconds.

Decentralized Environmental Notification Message (DENM)

DENM messages are not sent periodically as the CAM ones as it is explained on ETSI TS 102 637-3 [15]. These types of messages are triggered by an RHW event with the intention of providing information to other vehicles about new drive event or situation, described on table 2.3. The ITS device must be able to show to the driver the information in a Human to Machine Interface (HMI) of the receiving DENM, in order to use this information to take action in his driving.

In accordance with [15] the DENMs contains the following information:

- **ITS Protocol Data Unit (PDU) header:**
  - Protocol Version: indicates the current version of the protocol being used at the management container level.
  - Message ID : message type identifier associated to DENM
  - Generation time: time stamp when the DENM is generated in Coordinated Universal Time (UTC)
- **Management:**
  - Originator ID:ITS station identifier.
  - Sequence Number: sequence number provided by the originator when an event is detected for the first time.
  - Data version: indication an update of the event evolution.
  - Expiration time: in UTC.
  - Frequency: transmission frequency of DENM as defined by the originator ITS station.
  - Reliability: probability for the event information to be true.
  - IsNegation: negates the existence of the event.
- **Situation:**

- CauseCode: identifier of the event in a direct cause.
- SubCaseCode: value for more definition of the cause.
- Severity: severity value of the event.

- **Location container:**
  - Ref Position: Situation Latitude: latitude of the event reference position.
  - Ref Position: Situation Longitude: longitude of the event reference position.
  - Ref Position: Situation Altitude: altitude of the event reference position.
  - Accuracy: event position accuracy.
  - Other DEs and DFs for the relevance area and the location referencing: this block is defined and specified by the RHW application with variable sizes.

According to [14] the use cases, triggering conditions and terminations are shown in the following table 2.3:

Use case	**Triggering condition**	**Termination condition**
Emergency electronic brake light	Hard breaking of a vehicle	Automatic after the expiry time
Wrong way driving warning	Detection of a wrong way driving by the vehicle being in wrong driving direction	Vehicle being in the wrong way has left the road section
Stationary vehicle - accident	e-Call triggering	Vehicle involved in the accident is removed from the road
Stationary vehicle - vehicle problem	Detection of a vehicle breakdown or stationary vehicle with activated warnings	Vehicle is removed from or has left the road
Traffic condition warning	Traffic jam detection	End of traffic jam
Signal violation warning	Detection of a vehicle being violating a signal	Signal violation corrected by the vehicle
Road-work warning	Signalled by a fix or moving roadside ITS station	End of the roadwork
Collision risk warning	Detection of a turning collision risk by a roadside ITS station	Elimination of the collision risk
Hazardous location	Detection of a hazardous location	Automatic after the expiry time
Precipitation	Detection of a heavy rain or snow by a vehicle (activation of the windscreen wrappers)	Detection of the end of the heavy rain or snow situation
Road adhesion	Detection of a slippery road condition (ESP activation)	Detection of the end of the slippery road condition
Visibility	Detection of a low visibility condition (activation of some lights or antifog)	Detection of the end of the low visibility condition
Wind	Detection of a strong wind condition (stability control of the vehicle)	Detection of the end of the strong wind condition

Table 2.3: DENM use cases, triggering and termination conditions. *Source:* [14]

2.4 SMARTPHONES

Smartphones are totally introduced in the daily life of the people. This industry is constantly innovating and releasing new products to make everyone's daily life easier. These devices are also used to improve the VCN and ITS issues. It could be because they have become really powerful devices with a great hardware resources in a very small volume. Smartphones are equipped by powerful processors, high resolution displays, sensors such as, accelerometers, magnetometers, gyroscopes, location systems and high resolution front and back cameras. They have multiple types for a connection, for example, NFC, Bluetooth, Wi-Fi, LTE/UMTS/GSM or Universal Serial Bus (USB). These devices run a mobile oriented OS, that allow the development of applications that can take advantage of these features in a mobile context.

The smartphone market had been picking up in 2015 by the International Data Corporation (IDC) [16], growing 13.0% year over year, with 341.5 million shipments. The variety of smartphones is huge. Even if the number of manufacturers and different devices is significant, there are only two platforms that clearly dominate the market, those are Android and iOS.

In this dissertation the Android platform is selected as a personal AU of this VCN project. This decision is taken due to eCall++ is inheriting some functionalities from the Highway Environment Advanced Warning System (HEADWAY) [17] project and this projects AU was developed on Android. It is a matter of time until we see this application for iOS OS devices.

2.4.1 ANDROID

Android is an OS developed by Google that could be used in smartphones and tablets as well in embedded systems. It is based on a Linux Kernel. It was introduced in the market in 2007 as a results of an open-source project named Android Open Source Project (AOSP) from the Open Handset Alliance (OHA) led by Google. It was devoted to advance open standards for mobile devices.

Android offers to the developers tools and services to use in application design, development and distribution. Android developers website [18] offers to the developers some guidelines to application design to establish a standard for UX. Android has provided Android Studio for development since 2013. It is an Integrated Development Environment (IDE) that has all the necessary tools and information for application development. It is downloadable in Android developers website [18].

Android has Play Store application repository for the distribution. To publish an application in Play Store developers must pay a registration fee of 25 USD.

This OS also provides to the developers Software Development Kit (SDK). Though SDKs developers have access to big range of frameworks and Application Programming Interface (API)s. According to [19] the most significant SDK characteristics are the following ones:

- Access to device hardware, such as, camera, GPS and sensors.
- Data transfers using Bluetooth, NFC, Wi-Fi and USB.
- Background services.
- SQLite Data Base for data storage and recovery.
- Inter-app communication.
- 2D/3D graphics and media support.

- Geocoding and location-based services.
- Cloud to device messaging.
- Optimized memory and process management.

The principal programming language for Android application development is Java. To implement parts of application that require higher performance Android Studio also supports C and C++. Android Studio gives two different choices to run the Android application; running it on the device or in the emulator.

2.5 RELATED PROJECTS AND APPLICATIONS

As it is mentioned before, smartphones are really powerful devices. Their hardware and APIs help to the developer to create application in a huge range of possible areas. The transportation and traffic systems too are areas where some application have been trying to introduce these intelligent systems.

A research is made for the propose of knowing what application are nowadays available for the safety, comfort, information and efficiency for the drivers. The platform repositories offer to the user a great range of applications for the driving assistant for different mobile operating systems.

2.5.1 APPLICATIONS AVAILABLE IN PLATFORM REPOSITORIES

The result of this search presents some examples of relevant applications and projects that are related to this dissertation. This projects could be applications available online or projects from the scientific community that try to explore the smartphone capabilities in transportation systems.

In the repositories of Android and iOS it is possible to find a large number of applications for every category. The goal of this search is to assess if there are applications available that provide services in the three categories described in section 2.2.1 and if there is any application project who covers this dissertations goals. There are multitude of applications at Play Store and App Store that don't have a good service quality. In this research only the ones with good quality and a considerable number of users are selected.

The results are the following ones:

- **Automatic:**[20] this application works with an adapter connected to the OBD-II of the car and it takes advantage of this connection to provide services to the user. It provides human help in an emergency, shows the driving of the user in a real-time feedback, decodes car engine problems in the screen and locates your parked car. It is available for iOS and Android.
- **WerckCheck:** [21] is an auto accident checklist and mobile app that guides users through essential steps to consider a route. The app uses mobile device's location service, audio recorder and camera to document the accident. Although this application is presented by National Association of Insurance Commissioners [22] to help to claim the insurance.
- **Android Auto:** [23] this application connects the smartphone with the on-board computer to the car. This permits to use the mobile phone connection to the on-board computer. The

connection between these devices could be USB or Bluetooth. If the mobile phone has Internet connection the road information will be updated to the GPS navigation system of the on-board computer. This will help to save gas and do an efficient drive. This application also helps the driver to manager the phone calls and speak throughout the car system. It is only available for Android smartphones.

- **Peugeot Car Safe:** [24] it is a mobile application available for Andoid and iOS mobile phones. It is made for driving assistance in Peugeot cars. It provides to the driver a call assistance, vehicle localization, emergency button and car information services. This application has an emergency history to check all the incidents on driving.
- **WAZE:** [25] is a social drive application with a huge driver community supporting it. With it, the user could know the best route to the destination. The maps of the application are frequently updated. The locations of the traffic jams, accidents, radars, police checkpoints and stations with better rates, are described on it. The user is able to coordinate with other users to get any information from different locations. It is available for Android, iOS and Windows phones.
- **Porsche Car Connect:** [26] this application is designed for Porche users. It is developed for Android and iOS smartphones. This application gives to the user different services such as car controlling from the smartphone, mechanics information, driving statistics, car status, car finder and crash aid services.
- **WerkWatch:** [27] it offers to the user an automatic traffic accident detection and road notification services. This application uses the smartphones hardware for the accident detection and it is developed for Android and iPhone smartphones. It is provided by an Client/Server architecture by HTTP to inform about the state of the road.
- **SOSmart:** [28] this application is able to detect automatically an accident and send a notification to the selected contacts and associated institutions with the GPS position. SOSmart detects car accidents using the internal sensors, accelerometer and GPS. It also provides to the user a panic button. When it is pressed, an emergency notification with the location is sent to emergency contacts and associated institutions. The algorithm that is used has been designed using real car crash data from the National Highway Traffic Safety Administration. It has been developed for both Android and iOS.
- **iOnRoad Aumented driving:** [29] this driving aid application uses the augmented reality and smartphone hardware to warn about the roads dangers. It also uses the smartphone camera to work as a dash-cam. It is available for Android and iOS. The mobile phone could be connected to the on-board computer and it is possible to manage the smartphone from the car.
- **Collision Call:** [30] this application is provided by a mechanism that will call the emergency services in the country you are. This mechanism makes a call when the impact of the G-force becomes significant. It also informs to the contacts that have been selected on settings. It is available for Android and iOS phones.
- **CaroO pro (DashCam & OBD):** [31] this application is connected to the OBD-II from the car and it uses the smartphone to provide services to the driver. These services are video dash-cam, eco-driving, mapping, show car information, automatic crash detection, and it sends a Short Message Service (SMS) in case of accident.

- **Social Drive:** [32] is a social network where the user could upload a temporally warning to the platform and this warns the other users by a visual and sound about the incidents. The application is available for Android and iOs devices. It also provides to the user the right to verify or cancel the incident on the road.
- **SaveDrives:** [33] is an application available on the App Store, that provides to the user a help screen in the case that an accident happens. It is provided by a dash-cam service, map and possibility to browse other drivers on the road. The device only will know that an accident has happened executing an AAD that uses the accelerometer of the device. This application also generates a route history where the user could search the last routes, and recorded videos.
- **CarSafe:**[34] is an safety driving application. It combines different hardware resources such as, front and back cameras and others embedded sensors on the phone to detect unsafe driving conditions. The application is mainly used to monitor the drivers and road conditions and produce alert messages. It does not support accident detections.
- **ITS Connect:** [35] this systems is an ITS created by Toyota. The ITS Connect uses **v2i!** (**v2i!**) and V2V communications to provide to the driver information about the position of an Emergency Vehicle, the state of the next traffic light, or traffic information about the presence of vehicles and pedestrians in blind spots. It uses Japan's standardized ITS frequency of 760 MHz to receive and send information transmitted from external infrastructure and other vehicles. It is a test project already and there are only three cars build with this system.

The next table seeks to reflect a summary of the different projects and applications that are collected in this section. The specifications used to define these projects are the characteristics that eCall++ covers.

	Emergency Call System	AU	Passenger detection system	Road info	Connection to VCS	Video Camera service	Connection to OBU	Android	iOS	AAD	Black box
Automatic	✓	✓	✗	✓	✗	✗	✓	✓	✓	✗	✗
WreckCheck	✓	✓	✗	✓	✗	✗	✗	✓	✓	✗	✓
Android Auto	✗	✓	✗	✓	✗	✗	✓	✓	✗	✗	✗
Peugeot Car Safe	✓	✓	✗	✓	✗	✗	✓	✓	✓	✗	✓
WAZE	✗	✓	✗	✓	✗	✗	✗	✓	✓	✗	✗
Porsche Car connect	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓	✗
WerckWatch	✓	✓	✗	✗	✓	✗	✗	✓	✓	✓	✓
SOSmart	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓	✗
iOnRoad Aumented driving	✗	✓	✗	✓	✗	✗	✗	✓	✓	✗	✓
Collision Call	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓	✓
CaroO pro (DashCam & OBD)	✓	✓	✗	✗	✗	✓	✓	✓	✓	✗	✓
Social Drive	✗	✓	✗	✓	✓	✗	✓	✓	✓	✗	✗
SaveDrives	✗	✓	✗	✓	✓	✓	✓	✗	✓	✗	✗
CarSafe	✗	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗
ITS connect	✗	✓	✗	✓	✓	✗	✓	✗	✗	✗	✗

Table 2.4: Mobile application comparison

As the table 2.4 describes, some application satisfy the characteristics from section 2.2.1. There are also some that implement an AAD mechanism, video service and black box function.

However, the WerckWatch application and the ITS connect project are the only ones who are developed in VCN environment. The ITS connect project is also able to do V2I and V2V communications, but there is not any application that implements an eCall help request with an automatic passenger detection system. On the other hand, only the CaroO pro application is able to give an acceptable video service, but there are huge number of application available on line with dash cam service. Finally, there is no project or application that meet the needs of the eCall++.

In the following section, the available passenger detection systems are described with the intention of knowing exactly how the eCall ++ should detect the number of passengers in a vehicular context.

2.5.2 AVAILABLE PASSENGER DETECTION SYSTEMS

The car manufacturers and science community have been researching about passenger detection systems. At the same time, there is no application for the smartphones available for passenger detection. This search has been written to find out different systems, software or hardware, for passenger detection. These systems are not designed for smartphones but their development could be inspiring for this dissertation. The systems that are here presented, have been developed for researching and commercial systems.

- **Xerox Vehicle Passenger Detection:** [36] this system uses video from the outside of the vehicle to detect the number of occupants. The geometric algorithm identify vacant or occupied seats. In this system if the setting on the High Occupancy Toll (HOT) lane transponder does not match with the number of occupants, the system will take a snapshot of the vehicle's licence plate and alert law enforcement about the violator. It identifies the number of occupants in a vehicle with 95% accuracy, at speeds ranging from stop and go to 100 mph.
- **Delphi Passive Occupant Detection System (PODS):** [37] Delphi PODS systems have been installed on more than 36 million vehicles around the world. This system uses the pressure inside a fluid-filled bladder inside of the seat to classify the occupant. An electronic control unit processes the data and provides a deployment-allowed output to the vehicle's airbag controller when the defined threshold is met. This system is really easy to integrate into seat designs. Does not require driver action to suppress passenger airbag. It also provides optional support for seat belt reminder systems.
- **Nidec Occupancy Detection System (ODS):**[38] this system is composed of a single-layered, flexible sensor assembly that is thin, yet durable, and very easy to configure. The sensor assembly is installed in the bottom cushion of the passenger seat and is used to suppress or enable the passenger's airbag based on the classification of the occupant.
- **Video Occupant Detection for Airbag Deployment:** [39] images of the passenger seat are taken from a video camera mounted inside the vehicle to classify the seat as either empty, containing a rear-facing infant seat, or occupied. A monochrome video camera is used for this experiment. The system was automatically trained on a series of test images. Using a principle components nearest neighbour classifier, it achieved a correct classification rate of 99.5% on a test of 910 images.
- **Passenger Monitoring in Moving Bus Video:** [40] it is a person detection system for public transport buses tackling the problem of changing illumination conditions. It integrates a stable Scale Invariant Feature Transform (SIFT) background seat modelling mechanism with a human shape model into a weighted Bayesian framework to detect passengers on-board buses. SIFT background modelling extracts local stable features on the pre annotated background seat areas. It also tracks these features over time to build a global statistical background model for each seat. Since SIFT features are partially invariant to lighting, this background model can be used robustly to detect the seat occupancy status even under severe lighting changes. The human shape model further confirms the existence of a passenger when a seat is occupied.
- **Development of ODS Using Far-Infrared Ray (FIR) Camera:** [41] this project presents a system provide by FIR cameras to detect a human body. Also develops an algorithm for detecting automobile occupants, with the goal to reduce the risk of airbag-related injuries, develop smart airbag systems. This algorithm uses a detection of skin temperature range to determine physique and position of passenger.
- **Occupant Detection using Computer Vision:** [42] this project studies the different ways to detect an occupant in a car via computing vision. The Stereo vision, optical flow, HSV, Lines detection and face detection (NN, Motion and Colour functions) methods were used of this purpose. The final evaluation of this dissertation shows that all of these are good methods for passenger detection.

During this research two kind of passenger detection system are available, image detection system and special hardware (pressure/electronic) systems. The image detection systems uses a special processing for occupant detection. Normally they are equipped by a camera. The "Video Occupant Detection for Airbag Deployment" for example is able to detect an infant passenger. This system is installed inside of the car, but there are another kind of systems that are installed on the highway. The "Xerox Vehicle Passenger Detection" is one of those systems. It could be installed in the highway to detect if there is only one occupant in the car. If there is a car that drives with only one occupant in a Carpool lane it will be detected and the law enforcements will be alerted about the violator.

On the other hand, the pressure/electronic systems fit the seats with special hardware to detect the passenger number. This type of system is not the best for the eCall++. It adds more hardware to the system and it is not the intention for this dissertation to include external hardware to the smartphone for this functionality. It would be possible for the eCall++ to detect passenger with a communication to the vehicles equipped by OBD-II. Unfortunately, this connection is possible but the information is encoded and it changes depending to the automobile manufacturer. This closed information makes it really difficult to create this kind of passenger detection. It is better option to implement this functionality using the hardware from the smartphone. As in this section is seen the occupant detection by using image processing is possible.

2.6 CONCLUSION OF RELATED WORK

In this chapter, the principal knowledges were presented. The eCall chain to know the environment of this dissertation. The ITS concept and the VANET concepts were also explored. The smartphones were introduced too in the section 3.5, emphasising the Android OS. The goal of this dissertation is the development of an eCall application based on a smartphone with passenger detection and video streaming mechanism. A state of art of this kind of applications was researched in order to know the available application in this environment. Once knowing that there wasn't any application which detects passengers on a vehicle, a deeper research of passenger detection systems was done.

Once the background knowledge is described, it is possible to use this information to formulate a proposal. The next chapter 3 describes the proposal and aims to help to link the raw information gathered in this chapter to the actual application development.

CHAPTER 3

eCall++ architecture

According to 2.3 section, one of the most striking features of VC is that they are formed by OBUs and RSUs. In-vehicle domain the AU and the OBU must harmonize to provide an access to the user to the network. It is for this reason that the AU should implement an application that takes advantage of the OBUs communication capabilities. The AU also should provide of UI to permit to the users receive and send RHW messages, throughout the network. This type of unit could be integrated in the OBU or could be an external device, such as a smartphone, connected to the OBU. As it previously is described in this document, the eCall++ will work as an AU.

The goal of this dissertation is, on the one hand, to develop a smartphone based AAD and help request able to detect passengers, execute video streaming functionality in an VCN environment. The proposal aims to line up VANET requirements in order to develop a smartphone based application that could work as a solution for all requirements and still integrate them.

This chapter describes the proposal. First, eCall ++ architecture is described to understand how the application should work in an VANET environment. In this section the use cases of the application and requirements are depicted. Afterwords, the IT^{2}S platforms implementation is described in order to present how the OBU and RSU are built. The overall system integration describes how the connections between OBU and AU in the proposed architecture should be. Finally, the proposed implementation and integration are described.

3.1 eCall ++ architecture

As it was described in the previous chapter 2, the eCall++ scenario is based on the eCall and VANET architecture. Following figure 3.1 presents the eCall++ architecture. This architecture is inhered from Intelligent Cooperative Sensing for Improved Traffic Efficency (ICSI) [43] project where the *Instituto de Telecomunicações de Aveiro* was involved with other partners.

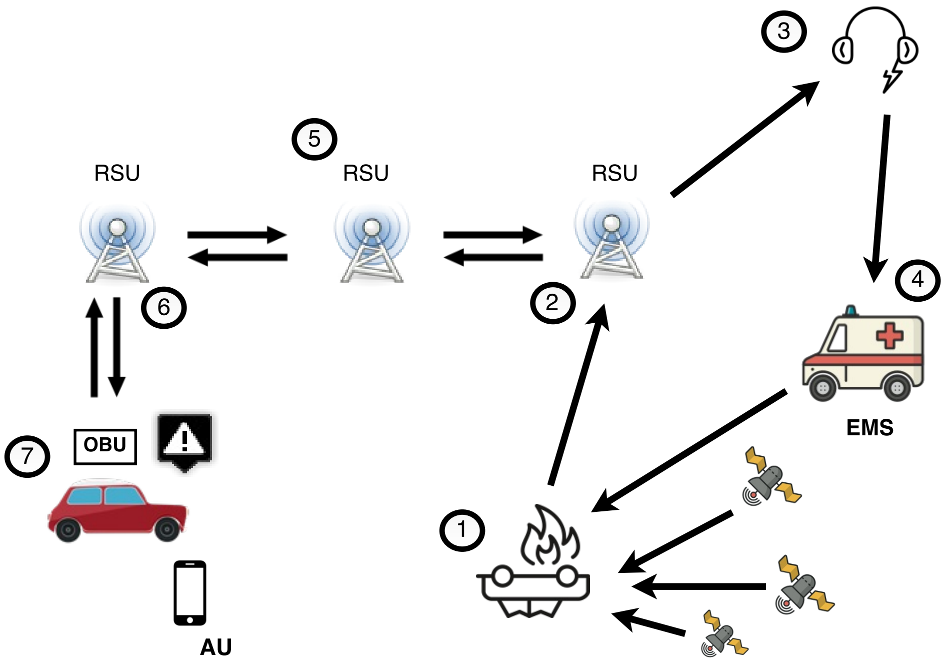

Figure 3.1: eCall++'s scenario

Having a look to the figure 3.1, if a car is involved in an accident, the eCall++ will detect it through the AAD mechanism ①. This mechanism should use the hardware from the smartphone to detect the accident. This help request will be inhered from the HEADWAY [17] and it is made to three ways. First an RHW data frame is sent to the OBU via USB. Second, consist to perform the eCall informing the EMS through OBU. During development of the HEADWAY project was verified that it was impossible to perform an eCall with the provided APIs. The third way, which was the solution for this problem, is to to send an SMS with the MSD data, followed by a voice call to the EMS. For this service the connection with the OBU is essential.

The OBU is made up of the IT^{2}S platform, depicted in the section 3.4. The MSD information, presented in 2.1, will be defined from inside of the car and from outside information, such as the GPS location. Once the OBU is informed it sends the RHW information, through an DENM message to the RSU installed on the road ②. The IT^{2}S platform has been design to work as OBU and RSU.

The RSU should send the accident information to the highway aid (number ③) and to the other RSU ⑤. On one hand, highway aid will send the EMS to the accident location ④. The MSD information received in the way is presented in the 4 chapter.

When a RSU receives new information it should inform the drivers on the highway. The OBUs of the cars will receive an RHW informing about the accident ⑥. The RHW are described on the 4.1 section.

Finally, the highway aid assistant could extract accident information from the video streaming mechanism of the eCall++ users ⑦. The user could also send a DENM message from the car to verify and inform about the accident. Furthermore, the smartphone and the IT^{2}S platform become a black box in case of accident. They store the recorded videos from the video streaming background mechanism, and photograph taken from the passenger detection mechanism.

The structure of this VANET is adopted from the ICSI [44] project and it contains parts from the IEEE protocol stack and parts from the ETSI protocol stack. These two protocol stacks are not completely disjointed. The following standards were implemented according to [45]:

- **IEEE P1609.0 - Draft Standard for Wireless Access in Vehicular Environments (WAVE):** specifies WAVE architecture and services necessary for multi channel DSRC/WAVE devices for vehicular environment.
- **IEEE 1609.1 - Trial Use Standard for WAVE:** describes the Resource Manager Application (RMA) services, interfaces and data formats for applications to communicate between architecture components.
- **IEEE 1609.2 - Trial Use Standard for WAVE:** defines the security services for applications and messages encryption in vehicular environments.
- **IEEE 1609.3 - Trial Use Standard for WAVE:** depicts network and transport layer services. It also define the WAVE Short Message Protocol (WSMP) and the Management Information Base (MIB) for the WAVE protocol stack. The WSMP is a network protocol developed for real-time communication.
- **IEEE 1609.4 - Trial Use Standard for WAVE:** Multi-Channel Operations and enhancements to the IEEE 802.11 Media Access Control (MAC) to support WAVE operations.
- **IEEE P1609.11 - Over-the-Air Data Exchange Protocol for ITS:** defines the services and secure message formats necessary for electronic tool payments.
- **IEEE Std 802.11p - This standard is an amendment to the IEEE 802.11:** defines rules and specifications for wireless local communications in 2.4 GHz, 3.6 GHz and 5 GHz, where the vehicular communication environment stands. IT^{2}S platform also supports the IEEE 802.11p who defines parameters such as transmit power limits, channel spacing and the frequency bands that may be used in each location.

This functional model presents a layered architecture conceived and divided in six main layers [44]: Application Layer, Network Layer, MAC Layer, PHY layer, Facilities layer and Security layer.

- **Application layer:** this layer is responsible for managing resources through the RMA and interacting with the OBU's resource command processor. It is based on the IEEE 1609.1 standard.
- **Network layer:** this layer defines the network transportation services. It includes the addressing data routing required for data exchange between WAVE entities. The data plane implements the IPv6 and WSMP protocols for short message exchange. This layer is based in IEEE 1909.1 standard.
- **MAC layer:** management plane named MAC Layer Management Entity (MLME) and a data plane compose this layer. The MLME is responsible for coordinating regular switching between Control CHannel (CCH) and Service CHannel (SCH) and queues service advertisements. The data plane is responsible for IPv6 and WSMP message queuing and transmission on the correct channel. This layer is based on IEEE 802.11p and IEEE 1609.4 standards.

- **Physical (PHY) layer:** it is responsible for data processing at the physical level it is responsible for encoding/decoding, interleaving/deinterleaving as well as modulation/demodulation operations. this layer is based on the IEEE 802.11p specifications for Orthogonal Frequency-Division Multiplexing (OFDM).
- **Facilities layer:** this layer is responsible for supporting a HMI for the driver, constant possession of location and time reference data, managing and crossing information from different sources, generating, routing and transmitting CAM and DENM messages, defined on 2.3.1. It is from ETSI protocol stack.
- **Security layer:** its responsible for security issues related to message encryption and validation. This layer is based on IEEE 1609.2 standard.

Once the eCall environment architecture is presented, in the next section the use cases of this architecture will be presented to describe the system behaviour.

3.2 REQUIREMENTS

The following requirements were mentioned directly or indirectly throughout the document. The main objective of this section is to summarize them all. The eCall ++ to be successfully integrated in a VANET should have some functional requirements to guarantee the performance, usability, safety and robustness. Below a summary of these requirements is presented:

- **Communication with the OBU:** a communication between the smartphone to the OBU is necessary to offer to the user the capabilities of the vehicular communication network.
- **Background execution:** this type of execution is necessary to run different types of mechanisms at the same time. In this dissertation the passenger detection,ADA and video streaming mechanisms are executed on background.
- **Use information and preferences persistent storage:** an constant storage is required to save the preferences of the user to not introduce those datas every time that the application is started.
- **Display information safely to the user:** the application should offer a combination of simple handling to the user to not distract the act of driving.
- **Accident Detection Algorithm:** an autonomous accident detection algorithm should be implemented in this application to inform as soon as possible to the EMS when an accident occurs.
- **Road Hazard Warning manual report:** the proposed application should offer to the user the way to send information from the road to the highway operator and other drivers.
- **Display OBD-II real time data:** the OBD-II connection offers to the application the opportunity of informing the driver about the car internal information.
- **Passenger detection mechanism:** a passenger detection is necessary to inform to the EMS about how many persons where in a car in case of accident.

- **Video recording mechanism:** this mechanism should be used in case that a vehicle is close to an accident on the road and the operator of the highway needs visual information about the incident.
- **Storage for dash-cam videos and inside car photograph:** a storage capacity is necessary to save the relative information about an accident, and consequently it might be possible to make a black box with a smartphone.

As soon as the eCall++ starts, the communication session should be established and the information is exchanged between the AU and the OBU. eCall++ must be able to work in background, while other application can work in foreground. eCall++ also, must be capable to save preferences, videos and photographs in the background. The exchanged data frames with the OBU should be defined before sending. The application should be able to automatically display the incoming RHW from the network with out distracting or visually entertaining the driver.

The ADA, passenger detection and video streaming services require access to the device's sensors, location and camera. In the ADA algorithm the necessary information must be transmitted when an accident is detected. eCall++ also should allow the driver to visualize vehicle's sensor data in real time. The OBU and RSU devices are used for this dissertation called IT^{2}S platform.

3.3 USE CASES OF ECALL++

After describing the eCall++ architecture, the eCalls's use cases diagram is presented to understand how the system interacts with the user as well as with other systems. To detail the eCall++ application software design Unifield Modeling Language (UML) has been used.

UML is a graphical language use for visualizing, constructing, specifying and documenting a system. UML diagrams should be used according to developer's necessity to specify and describe the software behaviour as well as its structure (static behaviour). To describe the eCall++ two diagrams are used, the Use case diagram and the Activity diagram. This last one is used in the chapter 4 to some implementation details.

Referring to the next use cases diagram, showed in figure 3.2, a first visual scheme of the application behaviour is defined.

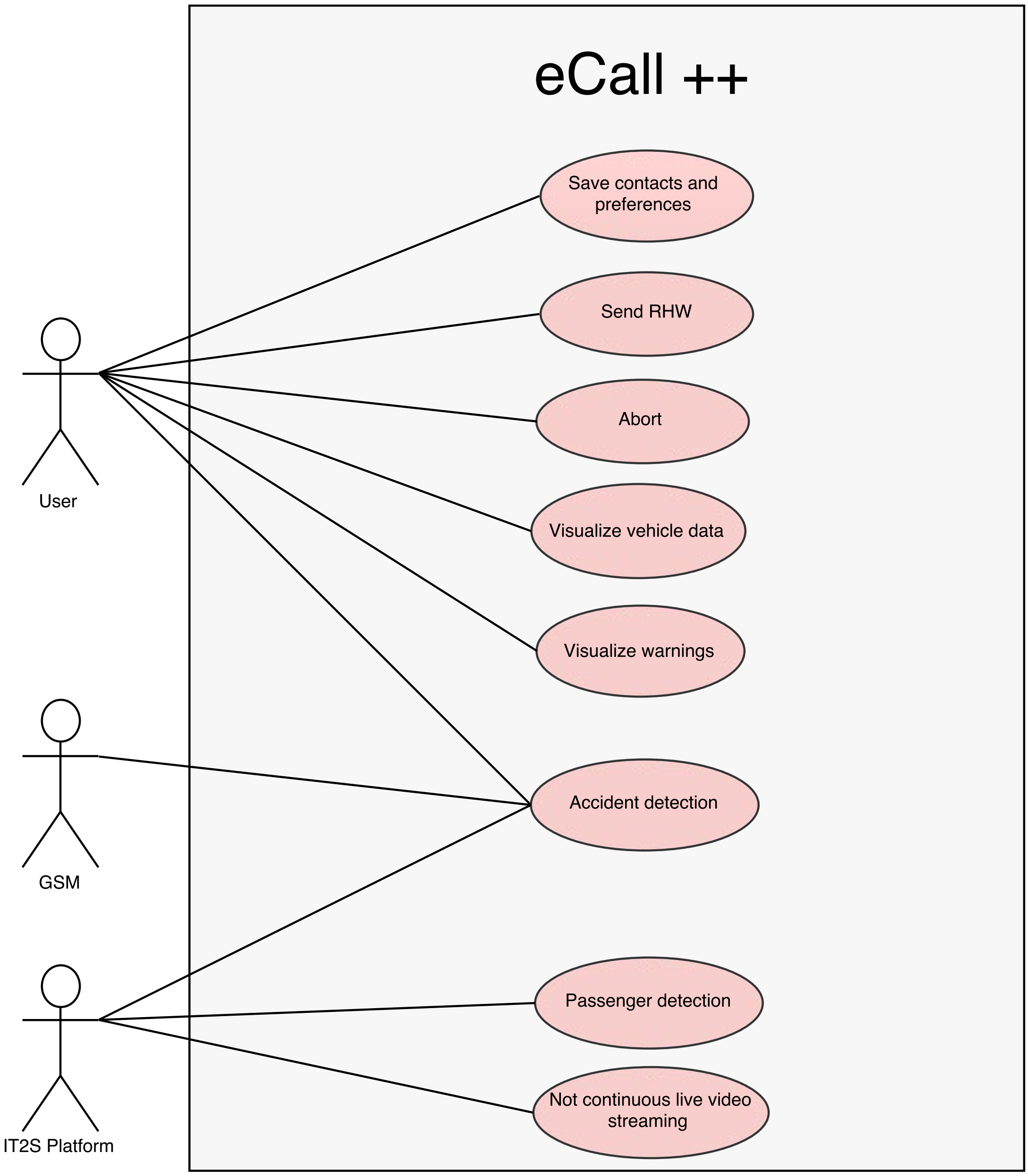

Figure 3.2: eCall ++'s use cases diagram

As the figure 3.2 shows, it is possible to identify three actors; the user, GSM network and IT^{2}S Platform. The red ellipses characterizes use cases. There are connections drawn between the actors and use cases of the eCall++. Connections represent the relation and interaction between both parts. The most important actor is the user, who uses most of the use cases.

Six use cases are connected with the user. Some of those use cases were implemented in HEADWAY [17] project, except the passenger detection, not continuous video streaming, which are use cases that the eCall++ is developing. The driver could interact with the application visualizing received RHW, through the Received activity. The driver also could reportRHWs pressing in the listview, through the Report Activity. The eCall++ should perform more cases of report and receives more RHW than the HEADWAY project. The user could visualize vehicle sensor data, through OBD-II activity. Also should be able to set and save the contacts information and preferences, by means of the UserSettings activity. There is an aborting help request option performed by the ADA. This is the way that the driver interacts with the ADA.

As it is previously commented in the chapter 2, eCall++ uses some functionalities from the HEADWAY project. With the goal to differentiate the work developed in each project the following classification was made.

In HEADWAY [17] project the user could implement the next use cases:

- Save preferences and contacts information, via the UserSettings activity.
- Report RHW Traffic, Hazardous Location and Accedent messages tabbing a software button, through the Report Activity in the previous version.
- Visualization of the RHW received, through the Received activity.
- Visualization of the vehicle sensor data, through OBD-II activity.
- Abort the help request performed by the ADA

In the eCall++ user could implement the previous ones and the following use cases:

- Passenger detection: passenger detection on the car, via a photograph taken in the Passenger detection Activity.
- Not continuous live video streaming: background video recording, storing and streaming.
- Send RHW: Report RHW cases, trough the Report Activity. In the HEADWAY the user only could send three cases, Traffic, Hazardous Location and Accident. In this new version the user could send seven new cases. Described on section 3.6.1.

As it is explained in section 4.1, RHW defines different types of cases characterised by Case and SubCase. This permits to define the event with higher precision. The Cases and SubCases used in the eCall++ are temporary, these could be changed or could be added more to the list.

In the use of case of ADA the user is the only one who can cancel the emergency call during the countdown, aborting the accident validation, however, the user is not capable to interact with the ADA directly. The OBU and the GSM network are the only ones who interact with the ADA directly. The ADA uses OBU to receive and send RHW messages. The GSM network uses the OBU to send SMSs to the contacts that were previously selected. These SMSs include the number of the passengers that is calculated in the passenger detection use of case.

The new use cases that the eCall presents are the passenger detection and non continuous live video streaming. The passenger detection mechanism is used by the OBU. The eCall++ is able to calculate how many passengers are in the vehicle by taking photographs with the camera of the smartphone device.

Upon an accident, the authorities or the EMS may ask a vehicle in the vicinity of the accident to share its smartphone video camera, in order to have a clear view of the accident site.

The use cases diagram 3.2 explains how the eCall++ behaves at low level. For a deeper perspective an Activity diagram was elaborated in the figure 4.4 in chapter 4. The Activity Diagram explains how is the application workflow in a higher level than the use case diagram. This diagram represents graphically eCall++'s work flow, i.e, the procedures resulting from the system functionalities and its interaction with the actors.

3.4 IT^{2}S platform as an OBU and RSU

IT^{2}S platform is a part of the structure defined in this dissertation. This platform is design to work as OBU/RSU for the under development system where the eCall++ will be integrated. This permits to the platform more flexibility to quickly integrate new and upcoming specifications. In this section, the IT^{2}S platform architecture is depicted to understand how the presented VANET works with the eCall++. It has been developed in the *Instituto de Telecomunicações* from *Universidade de Aveiro* and it has been used in some projects such as HEADWAY and ICSI.

3.4.1 IT^{2}S platform architecture

The IT^{2}S platform board is composed by the following main components [46]: Field-Programmable Gate Array (FPGA) module, a dual Radio Frequency (RF) fronted module and complementary electronic such as, Analog to Digital Converter (ADC)s and Digital Analog converter (DAC)s. The IT^{2}S board has numerous hardware interfaces to communicate and integrate with V2X communication architecture components. This board is developed with the purpose to allow its use as "Dedicated Short Range Communications (DSRC) - USB pen" on an embedded Linux computer. The architecture of IT^{2}S platform is defined in figure 3.3.

As it has been mentioned in section 3.1, the architecture has an implementation structure based on layers that requires different resources and therefore their implementation support can differ. In lower layers, PHY and lower MAC, more time critical operations such as synchronization, channel coordination and bit oriented operations are performed. The reason for this is that these functionalities are more efficiently implemented in hardware. The resource requirements by the Application, Network, Facilities, Security and upper MAC layers are implemented in software on embedded Linux computer. This is because these layers are less intensive than the lower layers.

The PHY and lower MAC layers are implemented in hardware. The digital PHY and lower MAC are implemented in an FPGA for flexibility purposes. The analog PHY, however, uses ADCs, DACs and a dual RF front end module from FPGA for transceiver, amplifier and RF switching implementation. The software and hardware implemented layers are connected through USB for data exchange and hardware upgrades.

An antenna connector is used to plug the DSRC 5.9 GHz antenna. This antenna is used for wireless communication between this type of devices (IT^{2}S platforms) using the IEEE 802.11p. The integration with the vehicle is throughout OBD-II interface. This connection is used to access vehicle sensor informations like speed, status error codes, etc.

At the moment the embedded Linux computer in use is a Raspberry Pi. This micro processor allows USB and Bluetooth connectivity. Figure 3.3 depicts that the IT^{2}S platform already expects smartphone integration to provide a GUI. The smartphone should be connected with the embedded Linux computer through USB or Bluetooth connectivity.

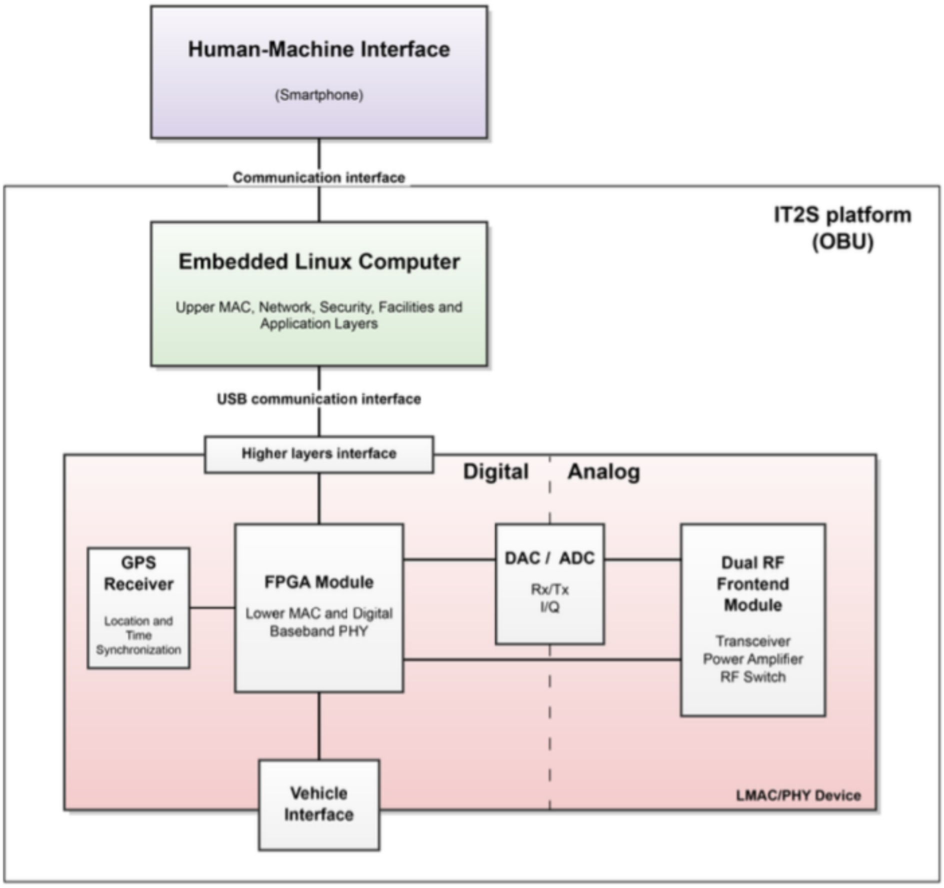

Figure 3.3: ITS platform architecture. *Adapted from:* [47]

3.5 SMARTPHONE AS AN APPLICATION UNIT

The IT^{2}S platform is equipped with the eCall++. This mobile application as an AU takes advance of V2X communications and eCall services. The eCall++ also adds passenger detection and video streaming functionalities to the VANET presented on 3.1.

This architecture implements the facilities layer from the ETSI protocol stack. This layer generates a well defined type of CAMs and DENMs messages with the intention to transmit information throughout the V2X communications. These messages are backbone of V2X communications. AU must be able to provide information to IT^{2}S platform to generate the CAM and DENM messages. Therefore, the AU should be empowered to receive these messages and show them to the user.

The DENM are messages generated of the user interaction and these are triggered by particular events named RHWs. These messages have directly impact on the road safety and efficiency. There are several predicted RHW use cases that are carried by DENMs, as the table 2.3 presents. eCall++ should provide the resources to report these exemplified RHW use cases. Some use cases can be easily detected by the driver, traffic condition, hazardous locations, stationary vehicle-vehicle problem, for example. Others such as, stationary vehicle accident should be triggered by an eCall system.

eCall++ uses the smartphone hardware resources to develop an AU. The large screen in smartphone allows them to display a GUI, that can be used as HMI for VANET, and present/report to/by the driver

RHW events. This project uses the smartphones several hardware resources, such as accelerometer and gyroscopes, to add the AAD mechanism.

The smartphones also are provided with front and back cameras. These are excellent hardware resources to collect information from road environment. The eCall++ uses these resources to detect the number of passenger and to stream video from the device. In this manner, saving this type of information in the device, it becomes a black box. As a result of the GSM connectivity of the smartphone the eCall help request also could be implemented.

The AU integration with the OBU also is an important part of the eCal++. This USB communication is inhered from the HDy Copilot project [48] that was part of HEADWAY. The Application layer is executed in the embedded Linux Computer. This device is the responsible for the communication and connection with the smartphone device. The AU as an extension of the Application layer is directly related to the OBU and they should be tightly integrated with it. The communication session should be initiated both, on the smartphone and the OBU's Application layer. Data formats also should be well specified in both sides. AU is responsible for detection of RHW and provides the related data to the OBU. Following the OBU will process this information and generate the DENM messages.

The next figure 3.4 depicts the purposed integration.

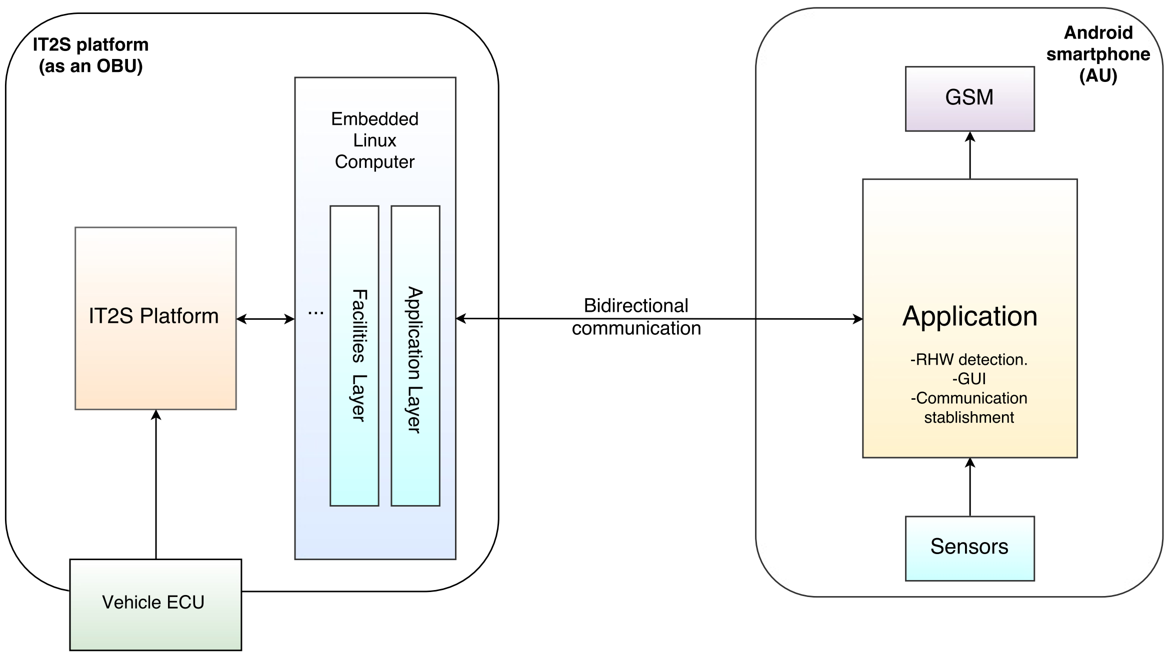

Figure 3.4: Overall system integration target implementation

The IT^{2}S platform is the support for the VANET. The smartphone is supposed to be integrated with IT^{2}S. This section presents a smartphone application to be integrated with the IT^{2}S platform. The application should act as a HMI that implements a GUI to report and pop up RHW messages. Another proposed issues to be implemented in the application are the ADA, passenger detection and video streaming mechanisms.

3.6 MAIN GUIDELINES OF THE APPLICATION

The guidelines presented in this section, aim to provide a link from the background knowledge to the actual solution proposal. To deliver on the dissertation objectives, eCall++ should provide the means to receive/transmit RHWs from/to other vehicles connected to the proposed architecture. RHW should be detected manually by the user, with a view to demonstrate that on-demand DENM messages are exchanged properly. The rich hardware resources, present in smartphones, should be used to develop an eCall system that would detect accidents, send help request and generate a RHW to be transmitted through this network. The RHW messages should take the passenger detection information from the passenger detection mechanism. The video streaming mechanism should be able to run in the background when the eCall++ application is running.

To accomplish these goals, four features are proposed:

- **GUI:** user interface to allow direct interaction with proposed network, incoming RHW information display and mean for the driver to manually report RHWs.
- **eCall system:** means to detect accidents automatically (AAD), help request such as eCall (call and data) and generate a RHW to be transmitted to the OBU (IT^{2}S platform).
- **Passenger detection system:** the eCall++ must be able to detect the number of occupant inside the car to add this information for the RHW.
- **Video streaming system:** eCall++ during the execution must be able to send video from the point of view from the inside of the car to VANET.

This proposal refers to a smartphone based prototype application and it's guidelines are more thoroughly explained along this section.

3.6.1 GRAPHICAL USER INTERFACE

The GUI is an important part of the application because it acts as a HMI. The IT^{2}S platform is supposed to be used in a vehicle and therefore in a mobile environment. The design and UX must be easy to use, with a simple display and no distract the vehicle occupant during the route.

To provide a solid and contextualized UX, the interaction should mimic the real navigation systems commonly used. The GUI also should provide an easy access to the services deployed in the system. The displayed content must allow a quick interpretation of data, to prevent the loss of focus on driving. The interaction with services must be designed to require minimal interaction steps. Also due to its context of use, the interface should be responsive and reliable, so it can provide confidence to the driver and add value to the system.

The smartphone should be located in the vehicle in a similar way to navigation systems. The driver should use a car holder as a suitable solution to fix the smartphone. Through this the driver could change the position of the AU of his choice to have a more comfortable driving. Once the AU is placed the driver should be able to see the device to interpret data as well as easily interaction with the device.

The interaction with the system should consist in to display of incoming RHW and to report them. As this system is a proof of concept only seven RHW use cases will be implemented for manual detection. Three of these cases are inherited from HEADWAY and four new from the eCall ++. These use cases can be easily detected by the driver. The following use cases are implemented by the eCall++:

- **Traffic congestion** (inherited from HEADWAY)
- **Accident** (inherited from HEADWAY)
- **Hazardous driving conditions** (inherited from HEADWAY)
- **Aquaplaning** (eCall++)
- **Animals on roadway** (eCall++)
- **Broke down vehicles** (eCall++)
- **Visibility reduced** (eCall++)

The proposed use cases are defined by Case and SubCase for a better description. Those are defined in the table 4.1. These Cases and SubCases have been chosen from the standard ETSI TS 102 637-3 [15].

The eCall system composition is divided in two; the AAD and the help request. The eCall++ must be able to integrate with the VANET, allowing the AAD system to send DENM messages.

To implement an AAD, the ADA inherited from the HEADWAY is proposed. This algorithm detects car accidents by using the smartphone hardware resources mixed with OBU information. In the event that the AAD detects an accident the smartphone connectivity should be used to perform an eCall and generate the RHW to trigger a DENM to inform about the event to VANET.

Passenger detection functionality should provide the information of how many occupants are in the car. The GUI of the eCall++ must warn the user that this functionality is running. This information should be consulted when the RHWs are generated, and it also will be useful in the generation of the DENM messages.

The EMS will know through the eCall++ application how many passengers are in the car and depending on the number of occupants they could decide how much help must be sent to the accident location.

The video streaming service should provide to the eCall++ a double functionality. On the one hand, the user will be sending a video streaming all the time to the vehicular network. The highway controller would have the information of what is going on in every vehicle that is on the road. This will warn the motorway controllers about events and it will increase the security on the roads. On the other hand, the smartphone and OBU will became a black box in case of accident.

3.7 PROPOSED IMPLEMENTATION

This section describes the proposal of the use of a smartphone as an AU. The smartphone must use a GUI for the interaction. The GUI should display and report RHW and inform the driver about the on road warnings. The system also should be able to detect the number of passengers in the car. Sending video streaming is one of the goals of this AU too. With this function there will be more information to know about the accident and the road situation.

3.7.1 GUI mockup

As an important part for the safety of the vehicle occupants, the GUI of the eCall++ should allow a quick and easy interaction graphical design. The smartphone screen real estate, should be used intelligently to easily display the necessary information to the driver, considering its position while driving. The UI elements should be large size and with high contrast colours for easily perceived, working in the same way during the day or night. The GUI is the connection between driver and eCall++. The interaction between both actors in the application should be limited, in order to not distract the driver.

The GUI displays data to the driver and allows him to interact with eCall++. Still, this display should be minimal to avoid the loss of focus on the road. It is proposed that the IT^{2}S interacts with the user via eCall++ application by visual RHW that concerns him, reporting RHW and visualizing vehicle sensor data. These are the main interactions the user should have with the application.

GUI should have three different layouts. One for the incoming messages (Received), another for RHW report (Report) and finally one layout for the vehicle OBD-II vehicle data (OBD-II).

Each layout is described following:

- **Received:** in this layout the RHWs are displayed. The RHWs are received as DENM content and they should be showed as quickly as possible in the screen. This layout should perform this functionality without losing the interest of the driver on the road. For this end the DENM's should be showed through symbols and with minimum text.
- **Report:** to report RHW the user should press on a list, that displays the type of RHW to be reported.
- **OBD-II:** this layout displays the OBD-II data, with this information the user can check what is going on in the vehicle system.

For a better understanding of this proposal layout mockups of eCall++ are presented on figures 3.5, 3.6, 3.5 and 3.8.

The layout that is drawn in the figure 3.5 represents the Received layout. This is the design where the driver could see the DENMs information. As is described previously in section 2.3.1 the DENM messages contain multitude of information, in this Layout only is the relevant information that should be presented. The type of RHW and the location. The remaining information should be used to process data in the Facilities layer.

Observing to the figure 3.5, it describes the remaining distance and the type of RHW. The large square presented in yellow colour, alerts the user of an upcoming RHW down the route. Three options of colour could be displayed in the alert square:

- **Red:** in the case that some RHW is reported in a short distance.
- **Green:** in the case that the road is safe and there is no RHW to display.
- **Yellow:** in the case that some RHW is reported in a fairly long distance.

With this scheme (Figure 3.5) the driver could easily interpret information regarding of the upcoming RHW. The alert square helps the driver to easily understand the gravity of the alert, and the distance to the event. Text will describe the necessary information to know what type of event it is.

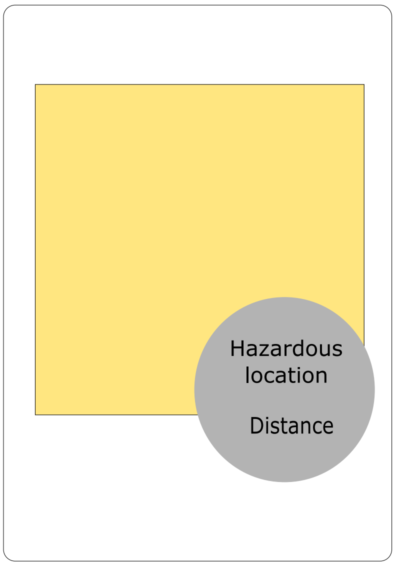

Figure 3.5: Received layout mockup

Figure 3.6 presents the Report layout. This layout provides to the user the functionality to report any of the seven RHW to the VANET system by tabbing in the text related to the event on the list-view. The list of events must be in a good text size with proper colours to not to distract to the driver. The background colour of the layout is white, in this way the driver will read it in any type of light. This or any other colour can be changed during development.

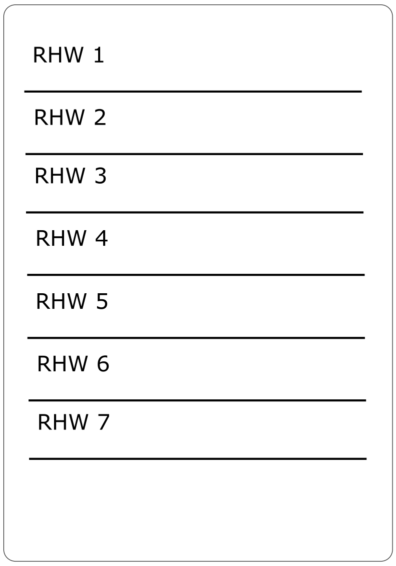

Figure 3.6: Report layout mockup

The GUI provides manual access to the driver, offering to the user the option to communicate with the network with RHW. Each RHW should be identified by unique code. Once the user manually tabs and reports a RHW, the device would be able to send the identification to OBU, so a DENM could be triggered from OBU. The next 3.7 figure describes the RHW manual report block diagram. The application has seven manual RHW as mentioned before. Each of these RHW should be triggered by pressing a dedicated GUI list.

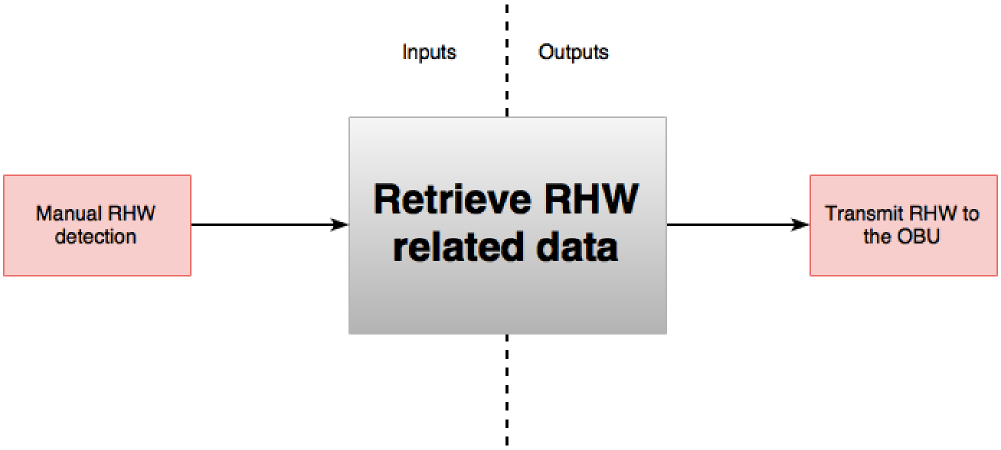

Figure 3.7: Manual RHW block diagram.

This feature is inherited from the HEADWAY but it should be updated to send the new information. This new information is the number of passengers inside the car. This is provided by the passenger detection mechanism. However, in eCall++, not like in the HEADWAY, the location data should not be sent to the OBU. The IT^{2}S platform will provide this information to the DENM.

The figure 3.8 presents the OBD-II layout's design. In this layout the information of the vehicle sensor is represented. This information is commonly used to detect vehicle problems. This information is displayed as a list view but in a different design than the Report layout. In this way, the user is not going to mix up with the Report layout. This list view describes each information with two texts. One for the title of the information and another for the data. These datas will vary the value during the driving time.

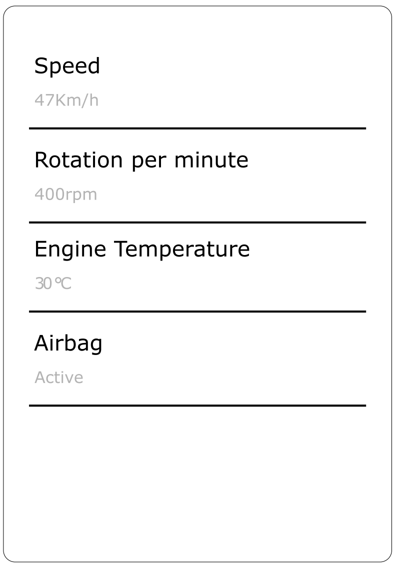

Figure 3.8: OBD-II layout mockup

The navigation between the precedent layout should be fast and intuitive, the user must be able to change from one layout to another naturally.

There are another layouts that will add more functionalities to the eCall++ application. One of those layouts is the settings represented on figure 3.9. The user could add relevant information and preferences for the use of the application, such as relevant emergency telephone numbers, other contact telephone numbers or the number of the countdown.

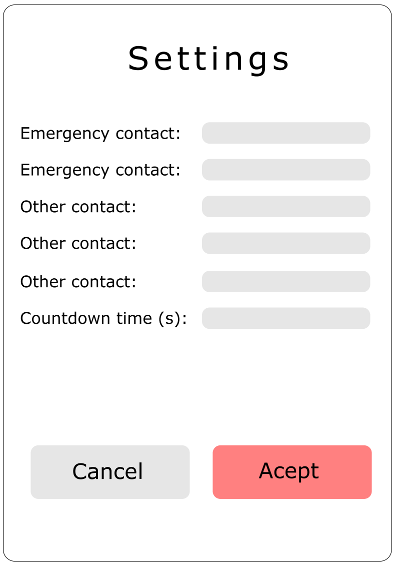

Figure 3.9: Settings layout mockup

Another layout is the passenger detection. This layout, depicted in figure 3.10, prepares to the occupants of the vehicle for the passenger detection mechanism. This mechanism should take a picture that will detect the number of passenger in the vehicle. This functionality is explained in the section 4.2.2.

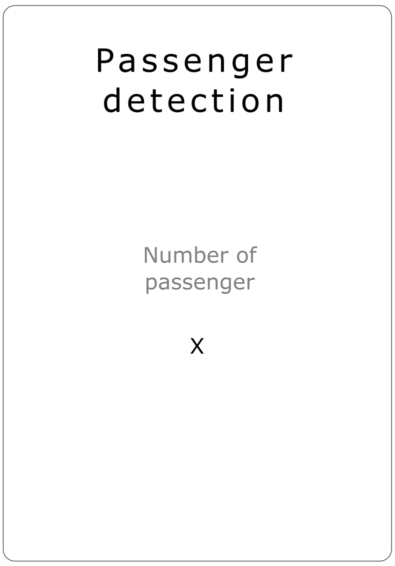

Figure 3.10: Passenger detection layout mockup

The mechanism that this layout executes, provides to eCall++ the number of occupants in the car. This mechanism should use the smartphone hardware and software to implement this functionality. So the passenger detection mechanism should use the camera from the device to take a picture and detect by image processing the number of persons in the picture.

This picture must be stored in the device making it a black box. The number of passengers should be stored into the device to provide the RHW with this information. This function only should run when the application starts, after this mechanism finishes the video streaming algorithm should start. The figure 3.11 depicts the passenger detection block diagram.

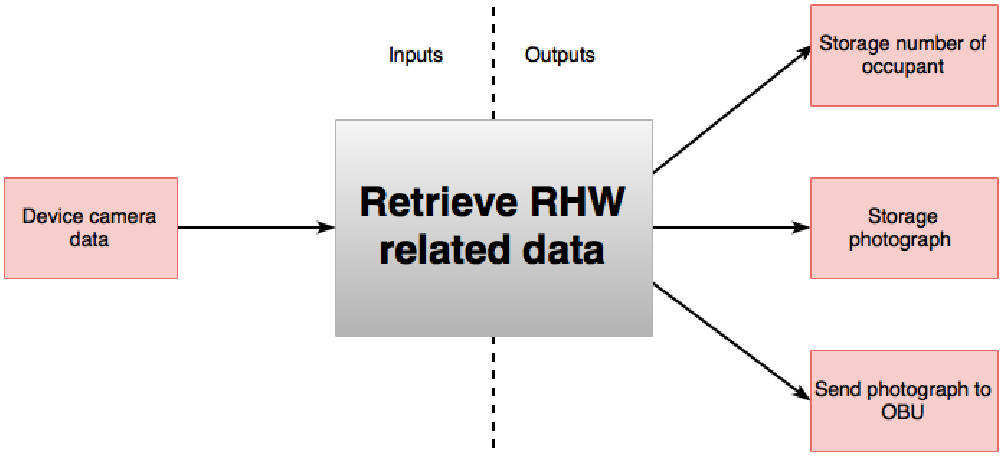

Figure 3.11: Passenger detection block diagram

The eCall++ during these layouts should implement new audiovisual functionalities to provide to the VANET new capacities. This mechanism is video streaming mechanism. The device should be recording and sending what is going on the roads from the drivers point of view. This video should be stored into the device, in order to use the smartphone as a black box. This video also must be send to the OBU, after this will send it to VANET. The block diagram of the sending video streaming mechanism is depicted on the figure 3.12.

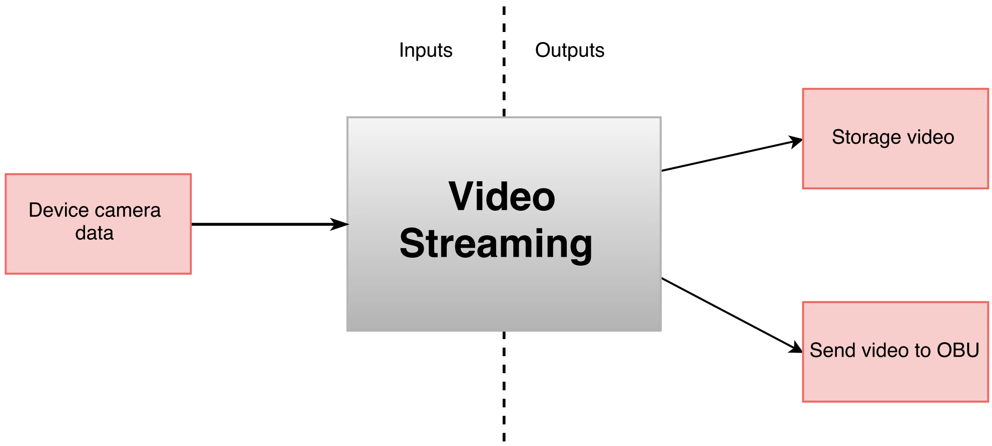

Figure 3.12: Sending video streaming block diagram

Having the proposal formulated, it is possible now to proceed to the development stage. The next chapter 4 describes with more details about the development process and choices.

CHAPTER 4

Implementation of the eCall++

Throughout this chapter, the implementation eCall++ will be presented, with the intention to introduce and describe the grouping of implementation that shapes this project.

The eCall++ was developed for Android 6.0, the most recent release of this OS. The IDE used for developing was Android Studio and the device used was a Samsung Galaxy Nexus, provided by the *Instituto de Telecomunicações* from *Aveiro*. This device possess all the hardware resources necessary for eCall++'s development.

4.1 Communication Interface

Observing the section 3.4, IT^{2}S platform is equipped with two types of connections, USB and Bluetooth. The Android device as an AU integrated in the IT^{2}S platform requires a continuous data exchange to execute all the goals of this project. There are four types of data exchange between the IT^{2}S platform and Android device; vehicle sensor data (OBD-II), RHW data, photograph of the interior of the car for passenger detection and video streaming service data.

The vehicle sensor data (OBD-II) are read from the vehicle CAN bus. The RHW can be transmitted in both directions: from the device to the IT^{2}S platform in an RHW manual report or by the ADA, and from device on the vehicle to the phone with an incoming RHW. The RHW data that is sent from the smartphone to the IT^{2}S platform and from the IT^{2}S platform to the smartphone is not the same, as presented in figures 4.2 and 4.3. Finally the passenger detection and video streaming mechanism exchange data in a unidirectional way. On passenger detection only one photograph is sent to the OBU. On the contrary, the video streaming mechanism is constantly sending the video from device. The IT^{2}S platform will cast what type of data is coming, video or photograph.

eCall++'s application main functionalities depend on the described data exchanges. The developed ADA in the HEADWAY [17] project, could use the vehicle airbag deployment signal to detect accidents. The communication between the main devices must be solid in order to guarantee a correct operation of the developed system. In view of these needs, the USB connection was chosen between the AU and

OBU. This type of connection provides energy to the device and ensures a good connection. This type of connection also eliminates the power usage problem.

Two of the previously mentioned data exchanges between smartphone and IT^{2}S platform, the OBD-II data and the RHW messages, are sent by fixed frames. Figure 4.1 describe the architecture of OBD-II data frames.

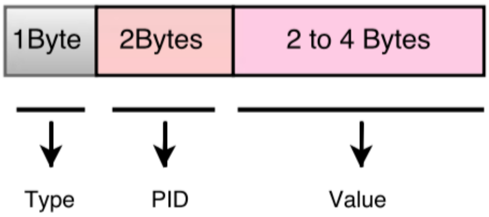

Figure 4.1: OBD-II data frame

As depicted in figure 4.1, the OBD-II data frame is composed by the Parameter Identificator (PID) and the value. The frame type's is one Byte. It signal the type of data carried by the frame. The OBD-II signal PID field is encoded by two Bytes and indicates the OBD-II signal (speed, airbag, etc.). This information is described at the figure 3.8. The next data field, the "Value", contains the value/quantity of the signal. This encoded signal varies from two to four bytes.

There are two types of frames for the RHW information. The RHW frame from the IT^{2}S platform to the smartphone informs the user about the state of the road. This is exposed in figure 4.2. This data frame contains a frame type indicator encoded with one Byte, followed by the RHW time stamp, latitude, longitude and type. The RHW data frame that is sent from the device to IT^{2}S platform is described on the figure 4.3. It is composed by one Byte with the number of passengers detected, eighth Bytes time stamp, one Byte type of RHW and one Byte from subtype of RHW. This RHW data frame adds the information of number of passengers. This information is one that the eCall proposes as optional information, as it is explained in 2.1.1. Actually these type of messages the RHW are the ones used to inform to the OBU when an accident is detected through the AAD algorithm. The eCall++ adds with passenger detection mechanism an automatic functionality that gives to the eCall this information without any manual manage.

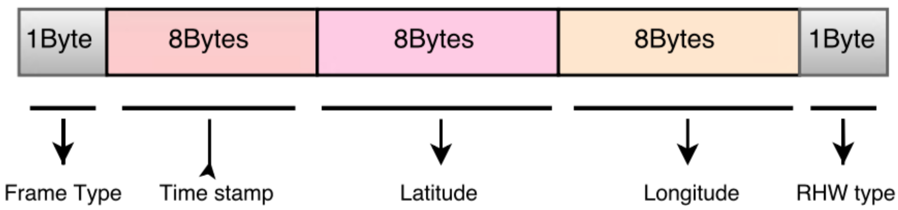

Figure 4.2: RHW data frame from the IT^{2}S platform to the smartphone

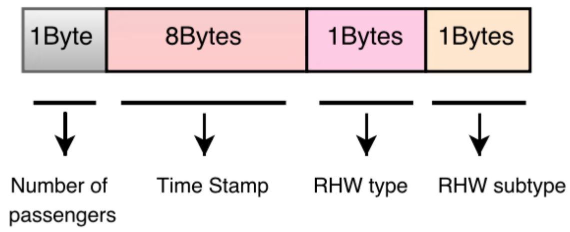

Figure 4.3: RHW data frame from the smartphone to the IT^{2}S platform

The RHW Cases and SubCases used on figure 4.3, have been chosen from the standard ETSI TS 102 637-3 [15]. These used Case and SubCases are the next ones:

ID	Name Case	Number Case	Name SubCase	Number SubCase
1	Traffic congestion	101	-	-
2	Accident	102	Multi-vehicle accident	1
25	Hazardous driving conditions	125	Rock falls	1
14	Aquaplaning	114	-	-
28	Animals on roadway	128	Large animal	4
31	Broke down vehicles	131	-	-
43	Visibility reduced	143	Visibility reduced due to swarms of insects	8

Table 4.1: RHW use cases in eCall++. *Source:* [15]

These RHW cases are not definitive. These Cases and SubCases are not chosen by the usability in a real scenario. In the HEADWAY project only the case number was exchanged between the AU and OBU, in this case Case and SubCases are defined.

4.2 ECALL++ IMPLEMENTATION

As previously described in section 4.1, Android platform was selected for the implementation of the eCall++. This decision implies that the Android guidelines [49] must be followed in the eCall++. These indications have been followed in the implementation of this dissertation and they could be divided in two:

- Architecture: Related to the application's functionality. It allows the GUI elements to perform tasks when demanded as well as other features that the developer introduces to the application.
- Design: Related to the GUI elements, such as icons, colours, layouts, images and visual effects.

The Architecture and Design are strongly related, but they are separately described and there could be some references between both.

4.2.1 ARCHITECTURE

This section is about the software design of eCall++. The design should be interpreted as the software architecture. It shows how different components should be developed and how they interact one with another. This section should not be interpreted as graphical design.

Following the software structure and behaviour are described. The eCall++ is targeted for Android OS, therefore, in this case the programming language is Java, which is object-oriented. However, the UML [50] was used to describe the eCall++ software design. Three diagrams compose the UML, that could be group as:

- **Behaviour diagrams:** it seeks to explain the system behaviour. To this group belong the Use Case, Activity, the State Machine and the Interaction diagrams.
- **Interaction diagrams:** these diagrams are subset from the preceding behaviour digram, that describes the interaction between objects. To this group belong Sequence, Communication, Interaction Overview and Timing diagrams.
- **Structure diagrams:** it describes the system elements that are independent from the time instant, such as the Class, the Composite Structure, the Object, the Component, the Deployment and the Package diagrams.

To describe eCall++'s software design three diagrams are used, the Use Cases in (section 3.3), the Activity and the Class diagram. These UML diagrams are accompanied by a textual description. The UML Activity diagram activities are distinct from Android activities.

BEHAVIOUR

Before starting to develop and write code it is necessary to design the project around the needs. This section describes the general functional schemes that have been designed to meet the specifications taken for this project. They are based on the eCall ++ use cases and requirements presented in section 3.3 and 3.2.

The Activity diagram is used to describe the eCall++. It is described in the figure 4.4. The Activity Diagram explains how is the application workflow in a higher level than the use case diagram depicted on section 3.3. The following diagram 4.4 represents graphically eCall++'s work flow, i.e, the procedures resulting from the system functionalities and its interaction with the actors.

As the diagram from the figure 4.4 shows the start of the USB session Activity comes first. The Android device is connected to the IT^{2}S platform though USB. This Activity is a background service running in a thread. If this connection is not possible the application displays a warning screen, defined in figure 4.18. Following, the application will be closed. If the session initiation succeeds, the application starts exchanging data with the IT^{2}S platform.

After this connection the Passenger detection activity comes. In this activity a photograph is taken from the front camera and the number of occupants is detected. It is noteworthy that in Android there is an activity responsible for launching the application. In this case StartActivity is the one that is described in the 4.14. The Passenger detection action has a rake symbol on its right hand side, which indicates that it has a sub Activity diagram. Sub Activity diagrams simplify the diagram, making it smaller and more understandable. This action's sub Activity diagram is depicted in figure 4.2.1.

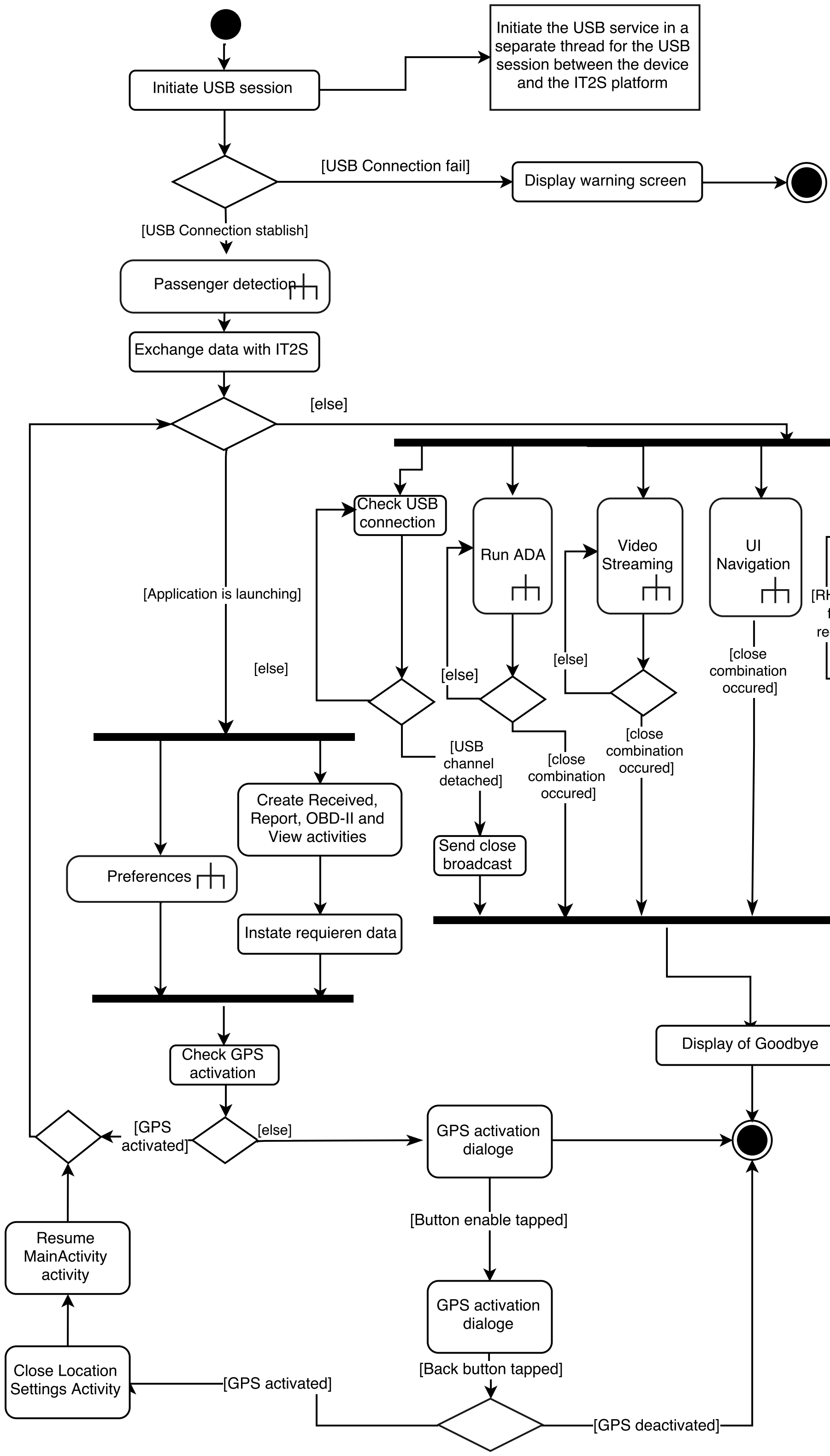

Figure 4.4: eCall++'s Activity diagram

As soon as the number of passengers is detected and stored in the device, the application sends the picture though USB to the IT^{2}S platform. When Passenger Detection finishes the MainActivity is launched.

When the application launches the MainActivity, it creates Received, Report and OBD-II activities. For that reason the eCall ++ instantiates some of the required data, such as, sensors data, location systems, etc. It also proceeds with the Preferences action. This activity is based on the same name activity of the HEADWAY project and is because of that is not described in this documentation.

The GPS location system is one of the tools that the devices possess. This provides an accurate location data to the eCall++ for the ADA and for calculating the distance between the user and an incoming RHW. In the HEADWAY project it was also used for RHW manual reports but in eCall++ this data is taken from the IT^{2}S platform. If the GPS is disable, the user is prompt with a dialogue box. The dialogue box is depicted on 4.16. There will be two options available for the user in this moment. On one hand, to press the "Enable" button, which leads to the Location Settings activity. This is an activity not belonging to the application. On the other hand, to push the "Quit" button, that is going to be followed by a goodbye screen, depicted in 4.17, which leads to the application termination. If the user is at the Location Settings activity, he can activate the GPS. When the user taps the devices back button to navigate back into eCall++, a verification is again made, and if the GPS is not activated the application will present the goodbye screen and terminate. If the GPS is enabled the Location Settings activity closes, the MainActivity is resumed and the application GUI is presented to the user.

At the present moment, eCall++ application launch state is over and it is not on the execution state. The "Run ADA" action also has a rake symbol on the right hand side, but this service is inherited from the HEADWAY [17] project and it is because of that that is not going to be explained in this dissertation.

The data exchanged between IT^{2}S platform and the device is constant. This information is formed by the RHW, the OBD-II signals and the videos recorded. The incoming data frames are constantly analysed. In the case a RHW data frame is received, the information will be displayed to the user in the ReceivedActivity described in the figure 4.10. The Display RHW action has a rake symbol on its right hand side.

Another important aspect of the application work flow, is the UI navigation. This interaction is bundled in the UI Navigation action. This sub Activity is depicted in figure 4.5.

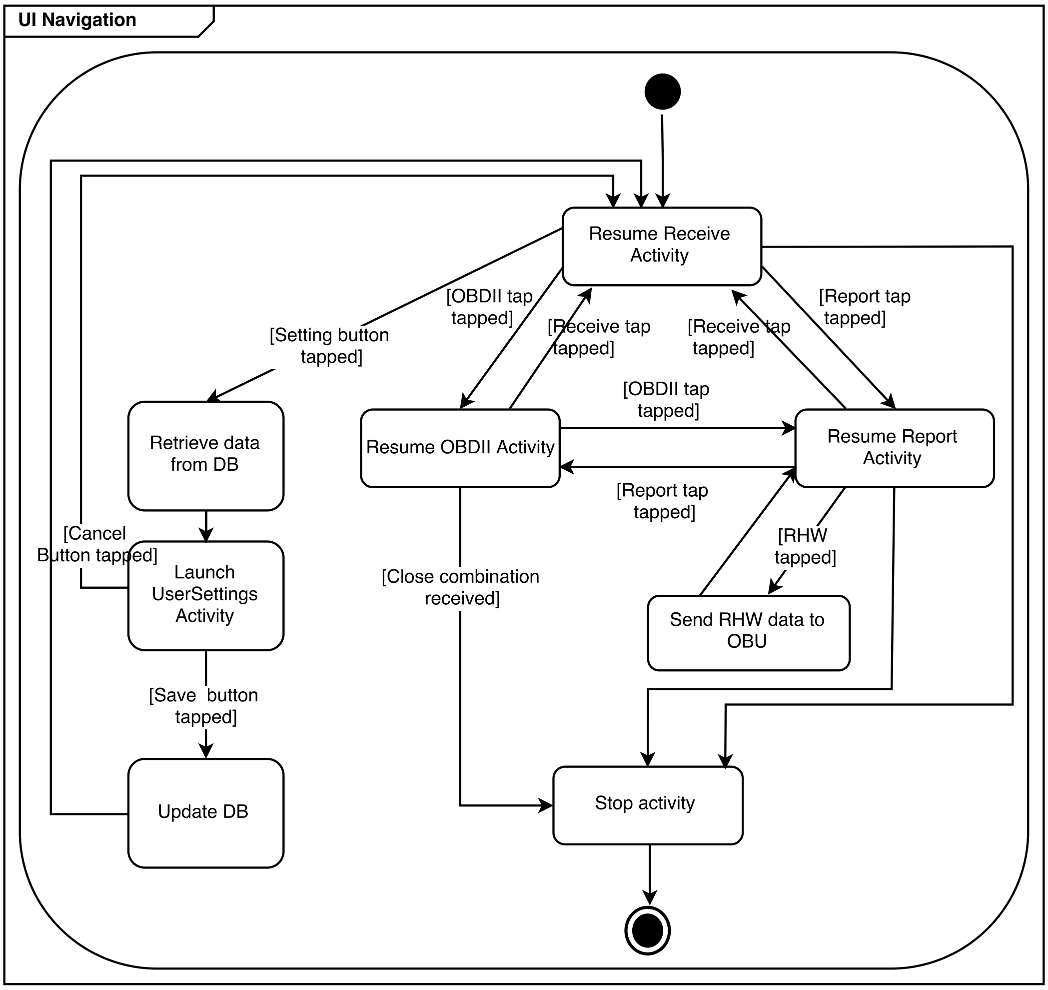

Figure 4.5: UI navigation sub Activity state machine. *Adapted from:* [48]

As the figure 4.5 depicts, the Received tab is automatically selected after the application finishes the passenger detection activity. The user, at this moment is able to navigate inside the application, using the tab navigator. This navigator has three tabs, as depicted in figures 4.10, 4.12 and 4.13. If the user taps the Report tab, Report activity is resumed. Here the layout depicted in figure 4.12 is presented to the user. This layout presents a list of textual description of the RHW they represent. When the user taps these items from the list, data such as, current number of passenger, type and subtype of the RHW is sent to the IT^{2}S platform. From here, the user can navigate back to the Received activity, by taping the Received tab, or he can navigate to the other activities. If the user taps OBD-II activity the layout depicted in figure 4.13 will be presented. Here the vehicle sensor information will be displayed to the user.

In this activity layout, as depicted in figure 4.5, is a Settings button. When it is tapped, the data from the Data Base (DB) is retrieved and the UserSettings activity is launched. Here the user could edit the preferences and contacts. When "Save" or "Cancel" buttons are tapped, the activity closes and the application returns to the Received activity, updating the DB in case the Save button was tapped.

The Android system itself generates broadcasts when certain situations occur, and in this case, the USBService broadcast receiver is ready to capture the USB cable detached broadcast. Once this broadcast is received, USBService generates a close broadcast to be received by all other activities. When the close broadcast is received by the activities, all execution related to USB data exchange are finished and the USBService is stopped. The ADA and not continuous video streaming service continue to be executed until the user taps the back button. If the back button is tapped, the goodbye screen is displayed and the application finishes.

The application is terminated when the USB cable is detached and the back button is pressed, respectively. This combination of events is the only way for the user to terminate the application. With this combination the ADA assures accident detection event even if, during a collision, the USB cable is detached.

As mentioned previously, the action named Display RHW has a rake symbol indicating a sub Activity diagram. This action is very important in the eCall++ and part of it is inherited from HEADWAY. The eCall++ updates this functionality and the RHW can be refreshed. The user can also receive new RHW notifications. As it is written in the chapter 3, RHW notifications should be displayed in a way that doesn't distract the drivers focus on the road. The Received activity layout was designed to allow a quick interpretation of the event. The RHW should be displayed to the driver with the minimum interaction with the device. The solution deployed is presented in the sub activity diagram depicted in figure 4.6.

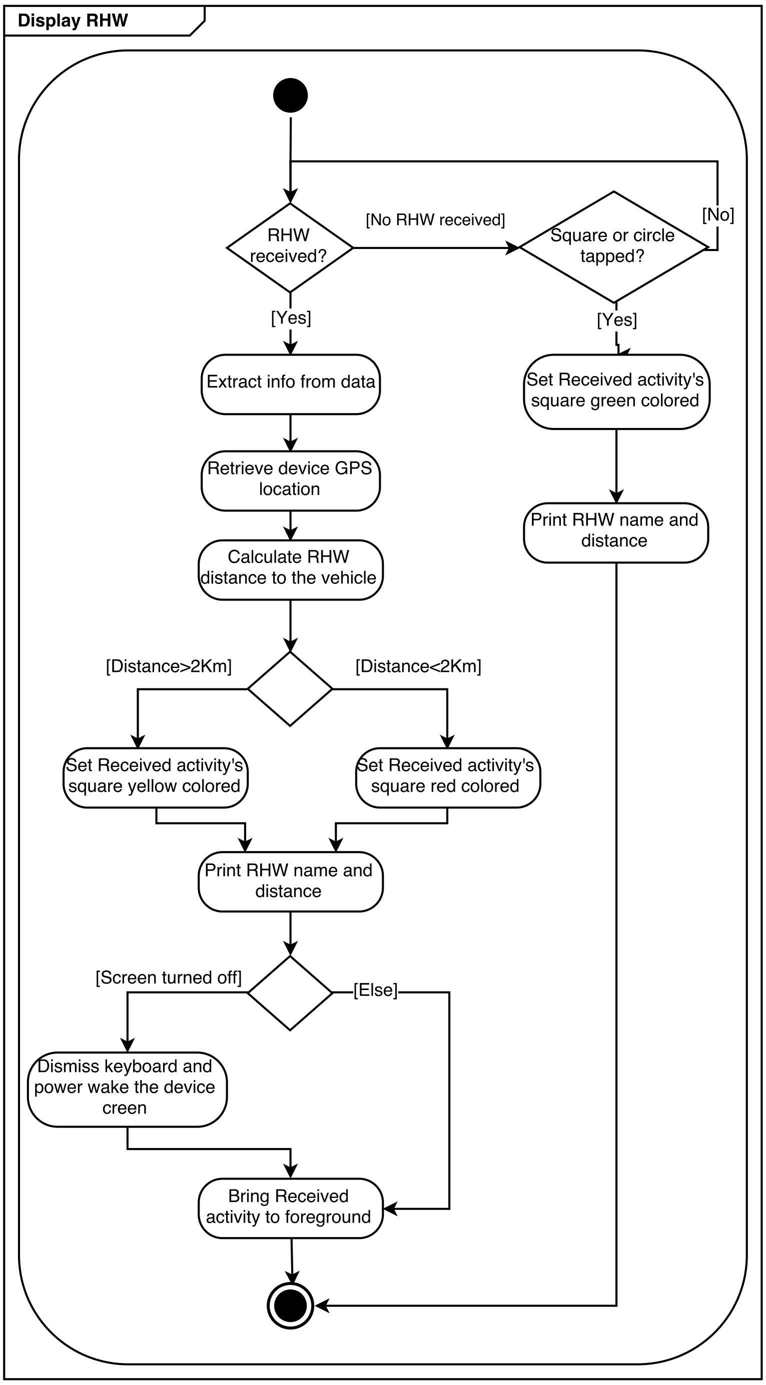

Figure 4.6: Display RHW action sub Activity diagram. *Adapted from:* [48]

As indicated in section 4.1, exchanged data is organized into specific data frames according to the information that they carry. Therefore, when a RHW data frame is detected it is displayed to the user. The RHW display process is explained in the figure 4.6. When a RHW data frame is received, its data, namely time stamp, location, type and subtype is extracted. Then, the devices GPS location is retrieved and the distance to the vehicle calculated. Depending on this distance the colour on the square from the display is updated. This value is merely for demonstration purposes and can be changed in the future. If the distance between the RHW notification location and the vehicle is more than two kilometres the square will be painted yellow. If this distance is less than this distance the square will be red. The distance and the type of RHW is printed on the grey circle of the Received activity layout, presented in the figure 3.5. After these alterations to the layout occur the application dismisses the keyguard and turns on the screen (if its turned off) and brings the Received activity to the foreground. The user then, only needs to have a look on the posted information to interpret the RHW.

In the case of there isn't any information received the user could refresh the state of the screen just tapping the information square or the circle of the layout. After the refresh if there is more RHW notification will be checked. The refresh is a functionality that just cleans the previous RHW events from the screen. The eCall++ should wait for a new RHW after the refresh.

There are still two last actions to be described in the eCall's Activity diagram. The passenger detection and not continuous video streaming which are fairly complex. Their implementation and sub Activity diagram are described in the following subsections.

PASSENGER DETECTION

The eCall++ Android application implements some use cases that the HEADWAY project didn't implement, one of those is passenger detection mechanism. The passenger detection is a mechanism that works at the first Activity, the StartActivity.

This mechanism is executed in the background, so the user of the application does not perceive that the smartphone is calculating the number of occupants. In this way, the user only is going to see an Activity on the screen. This Activity is described on the figure 4.14. After the number of passengers is computed the application will continue running as is described in the diagram 3.2.

Initially the goal of the eCall++ was to detect the vehicle occupants during the driving, but there are some technical problems for video streaming and this mechanism working in the same time. The problem is that it is impossible to have opened both cameras, front camera and back camera, from the smartphone at the same time. The cause is because the smartphone share the camera software resources for both cameras. This is a limitation for this type of smartphone application. In some applications the cameras are constantly switched, but the open/close time is that long that the application became really slow and useless.

Android SDK is used for this project. It contains an *FaceDetector* [51] API for face detection and the *android.media.FaceDetector* [52] class for this functionality. This API works for all the Android devices from the first version until the most current. This *FaceDetector* class is able to detect faces in an image. It uses the *findFaces* method to find those faces. *findFaces* method returns a number of detected faces and fills the *FaceDetector.Faces[]* array. The *findFaces* method supports only bitmaps *RGB-565* in format, so it is necessary to make a conversion from ".png" to bitmap.

The *FaceDetector.Faces[]* information array contains: confidence that it's actually a face (a float value between 0 and 1), distance between the eyes (in pixels), position (x, y) of the mid-point between the eyes, pose rotations (X, Y, Z). This API doesn't contain a framing rectangle that includes the detected face but a code which is developed for test in chapter 5 to detect false detections. The mid-point between the eyes data was used for this functionality.

The *android.media.FaceDetector* class is only able to detect faces, it does not support to implement face recognition functionality. The processing that is used for the detection of passenger in this project is explained hereafter in figure 4.7.

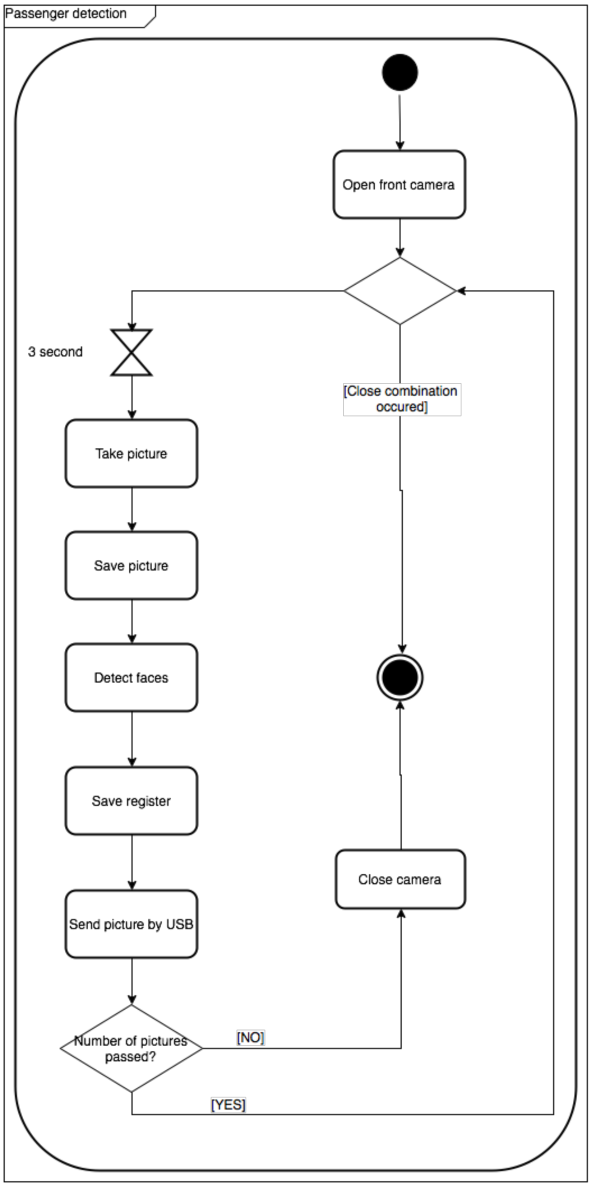

Figure 4.7: Passenger detection Activity diagram

As described in the figure 4.7, the first activity that is working is the one which opens the front camera. The mobile phone is placed in such a way that the front camera is facing at the inside of the car, where the passengers are sitting. It is really important to open and close the cameras before and after of using these resources. On many occasions some applications in Android OS are not able to use the camera because another application is using it in the background [53]. This functionality is working in the background and it can create some havoc to another applications. When the camera is open the device will check if it is equipped with this hardware. If is not the eCall++ will use this mechanism in this device.

A timer of 3 seconds is executed if any close combination does not occurred. This timer has two purposes. On the one hand, using this timer the user has more time to realise that a photograph is going to be taken. On the other hand, the time for taking a picture and saving takes 1 second and 23 milliseconds. This time is calculated by debugging and using time stamps of the beginning and finish of taking a picture.

Afterwords, the smartphone passes to the next activity and takes a picture from the inside of the car using the front camera. This action is developed using the *Camera* [54] class from Android. This class is available for all the versions of Android. This class provides to the developer multitude of functionalities, such as the autofocus. The *Camera* class is going to be replaced by *Camera2* [55] class but it is not able to work in the background yet, without using the default UI from Android.

For this action it's necessary to declare a *SurfaceView*. This permits the developer to display what is being seen from the camera. Android forces all applications to declare it because of privacy legality for users. In this way, Android avoids spy applications and the user knows that the application is running in the device using the camera. The point of this, is that there is no minimum size for the *SurfaceView* regulated. In this action there is no intention to distract the driver and because of that the *SurfaceView* size is of 1px per 1 px. The user is not going to perceive what the camera is focusing on. At the same time, a text describing the action is posted in the Activity to make sure to the user that the passenger detection is running.

Once the picture is taken, it will be stored in the smartphone. Only the last 5 photographs will be stored. In this way, the device is not going to be full of photographs. This also adds a new function to the smartphones as a black box. In case of an accident occurred the device could be analysed to know who were the passenger before the accident. After the photograph is stored, the *FaceDetector* class will detect the faces from the previously taken picture.

Once the number of faces is counted the number of passengers will be known. Then in the save register activity the number of passengers and the time stamp will be stored in the device. For this end, a Java class called *Register* was created. It has two parameters a *Time Stamp* and an *Int*. On one hand, the Time Stamp describes the exact time when the picture was taken, this means that the device could be a black box again describing the last time that the application was used. On the other hand, the *Int* data describes the number of faces that were detected on the picture.

Later, there is a number of taken pictures that should be taken for the passenger detection. This permits to select the maximum number of occupants detected and in case of accident that number is the one which will be selected to send in the RHW and on MSD. The number of taken pictures is variable and in the current version it only takes six pictures, but the algorithm is especially designed to be scalable.

Finally, the camera must be closed, not to bock smartphone's resources and not obstruct the following not continuous video streaming mechanism that is described bellow in the next section.

NON CONTINUOUS LIVE VIDEO STREAMING MECHANISM

Apart of the passenger detection mechanism described in the previous section, the eCall++ contains more use cases. The one that is presented in this section is the non continuous video streaming mechanism. The not continuous live video streaming mechanism is designed to send to the VANET administrator a video streaming to inform about the road. This mechanism is executed in the background. The user of the application does not perceive that the smartphone is sending the video from the camera at all. In this way, the user can use the application in the same time as the smartphone is running this service.

The non continuous video streaming algorithm requires the back camera from the smartphone. This Camera is used for the video streaming, and the devices which are not equipped with this hardware are not able to run this service.

The service is running in the background of the Receive, Report, OBD-II and View Activities. The video streaming sending service algorithm follows the next diagram activity 4.8:

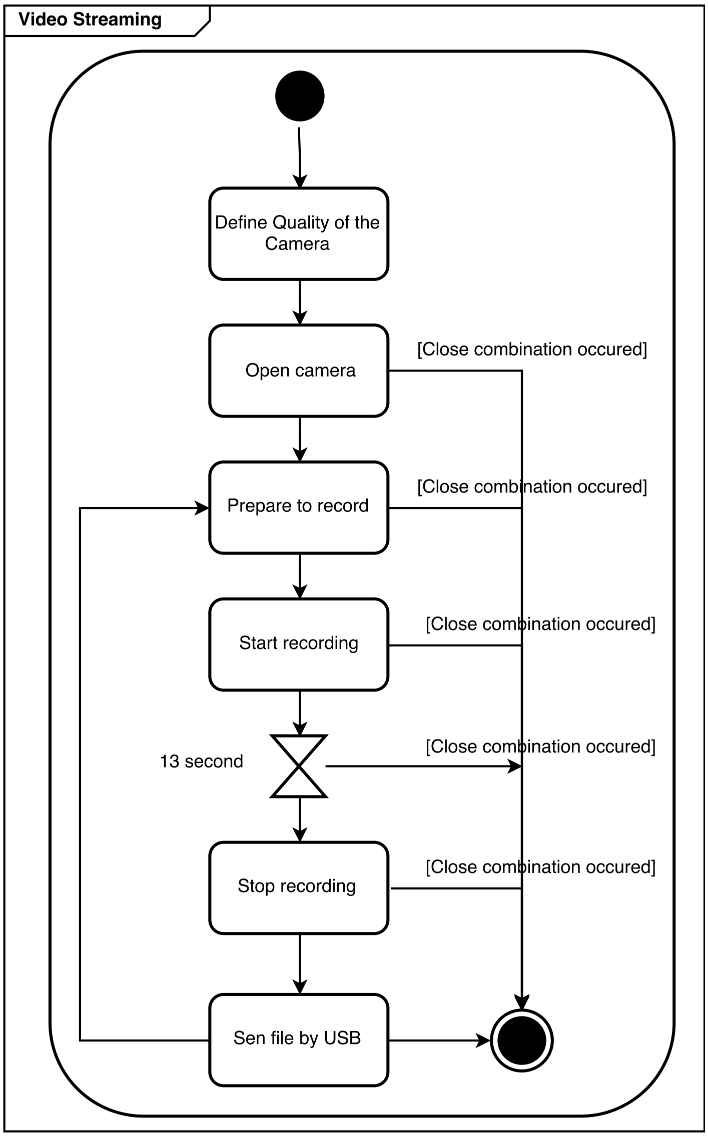

Figure 4.8: Not continuous live video streaming diagram

Observing the 4.8 diagram, the mechanism starts defining the quality of the video that will be recorded. This is caused because one of the classes that is used for this functionality is *CamcorderProfile*. This class defines the quality of the source camera. The quality defined for this camera is named *QUALITYHIGH*. This quality level is going to correspond to the highest available resolution from the camera.

Following, as in the same way as the passenger detection mechanism, the back camera is opened. Afterwords, the *MediaRecorder* class is defined on "Prepare to record". In *MediaRecorder* the quality of the audio and video sources are defined. The name of the future record video also is defined in this class. In this service the name of the video is always going to be the same. In this manner it allows to store only one video, and the capacity of the device is not going to be exceed. The *MediaRecorder* class also is able to define other characteristics for the video that is going to be recorded in this service, such as, maximum duration or maximum size in bytes. The *MediaRecorder* is linked with the *ViewSurface* of the MainActivity. This is induced because Android does not permits to record from the device if the user does not watch what is recording.

After these datas are defined, the device is prepared to record a video. the recording activity takes 13 seconds. After this time the recording is stopped and the smartphone sends the video to the IT^{2}S platform by USB. Following the *MediaRecorder* class must be defined again, and the " Prepare to record " is executed. This service could be stop at any time because a combination of functions occurred. This could be an accident detected by ADA or the shutdown of the USB connection.

This service needs 1 second and 4 milliseconds to store the video and send it by USB. It is because of that is that the streaming is not continuous. This is because there is not possibility to execute two recording threads at the same time in an Android device. Android only has one access to the Camera. It must be opened and closed to record a new video. The *MediaRecorder* class is used to this functionality and to store the video on the device. There is no possibility to use two *MediaRecorder* classes and access the hardware camera at the same time. This is the reason why 1 second 4 milliseconds is lost in the video streaming and, consequently, it is not a continuous service.

STRUCTURE

The eCall++'s dynamic behaviour is described by the Use Cases and the Activity diagrams from preceding sections. The goal of the structure is to explain the development of eCall++, having access to the static behaviour of the application. The static behaviour is presented in figure 4.9. The attributes and methods of the Java classes are hidden to simplify the diagram.

The eCall++ is composed by seven activities; StartActivity, MainActivity, ReportActivity, ReceiveActivity, OBDII, UserSettings and Countdown. The StartActivity launches the application and starts the passenger detection mechanism. Once this service is finished the MainActivity is displayed. This activity is responsible for launching most of the other activities as well as starting the service named USBService. This activity is also responsible for managing the application lifecycle.

The UserSettings activity is used to save preferences and contacts introduced by the user accessing to the database through its layout interface. The ReportActivity activity manages the RHW items from the ListView of the layout. Here, when a item is tapped, the activity gathers and organizes the RHW related data to be transmitted to the IT^{2}S platform.

The OBDII activity is used to display and analyse incoming vehicle sensor data. The Receive activity is responsible for displaying the RHW and the distance to the event to the user through the layout. This activity also is accountable of the ADA algorithm execution and the video streaming mechanism.

The CountDown activity is launched once an accident is detected by the ADA. The CountDown activity permits the user to assess the accident detection validity. Once the time of the countdown of the activity is over this activity does a help request. IT contacts with the emergency services by the GSM network and sending SMS to the contacts.

The USBService is a Java class that is used to manage the USB connection and its data change, such as, the RHW messages, photograph and the video streaming. This class inherits the attributes and properties of the Service superclass.

The rest of objects are also part of the eCall++ structure. Values object encapsulates the vehicle sensor data to be displayed in the OBDII activities's layout. This information is exposed as a list from the OBDII activity's layout. The DB object creates a SQLite database table. This database is accessed and manipulated by the UserDB object. The data to be stored or retrieved to/from the database is encapsulated by an object named User. The Waker object is responsible for dismissing the device key guard, powering the screen and bringing the Received or Countdown activity to foreground, if the user receives a RHW or if he suffers an accident respectively.

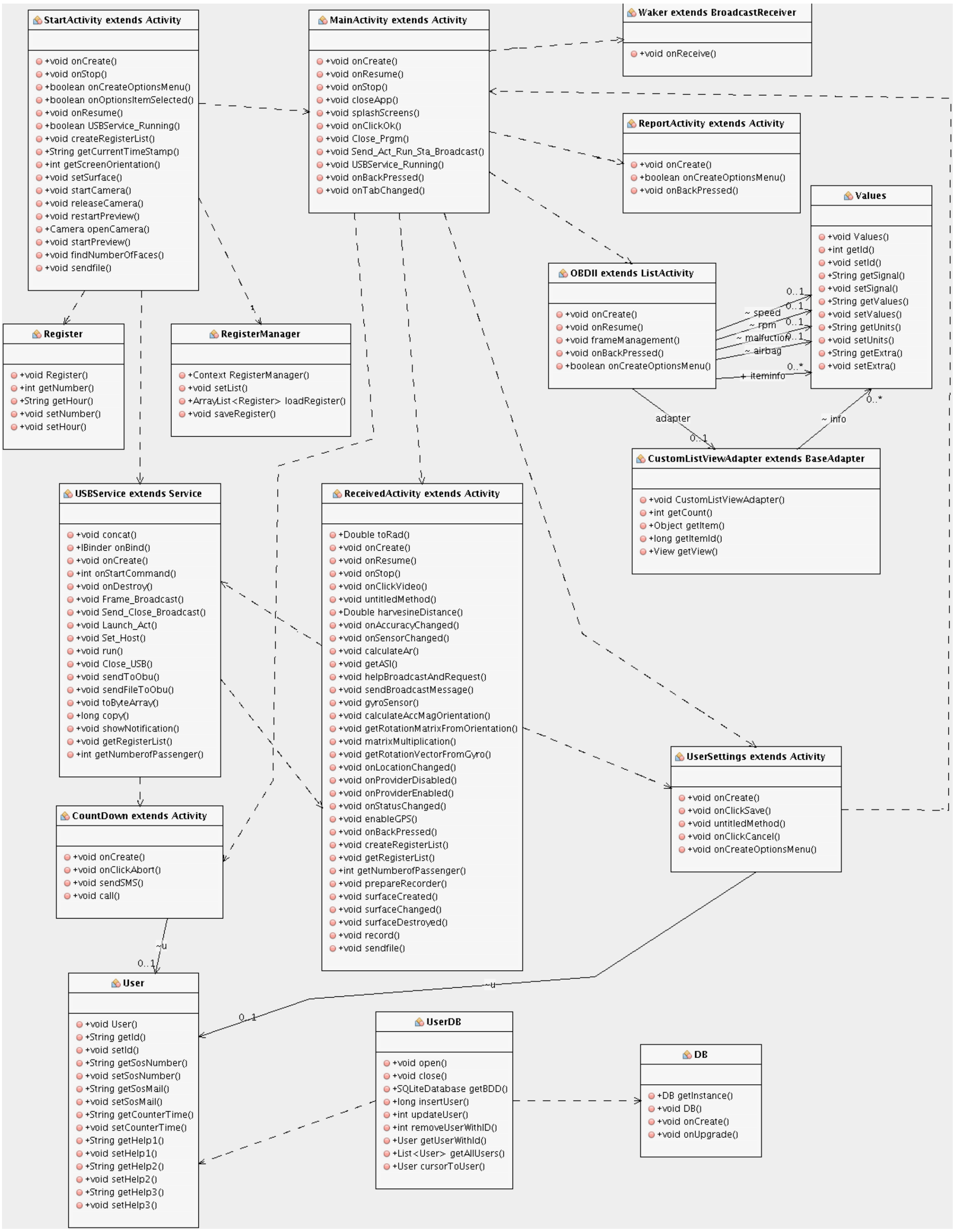

Figure 4.9: Class diagram

As it is mentioned previously the activities and services that communicate between each other through broadcast.

Once the architecture is defined in the next section the design of the application will be described.

4.2.2 Design

The GUI is an important part of the eCall++, it influences the interaction between application and user. As it is known, the Android was selected for this project. The design of Android applications varies from platform to platform.

The eCall++ has followed the GUI mockup for the design presented, as a proposal, in section 3.7. On top of that, the Android design guidelines are followed [49] to produce intuitive layouts for Android and non-Android users. The main goal of the design of the eCall++ GUI is the simple, easy handling and the superficial it is. This simple minimal touch interaction and basic draws causes less attention from the driver. The type of navigation implemented is tabs, this is recommended by the guidelines for superficial applications [49]. The main activity consists in three tabs, Receive, Report and OBD-II activities. With these three taps the driver could quickly manage the different activities and achieve his goals.

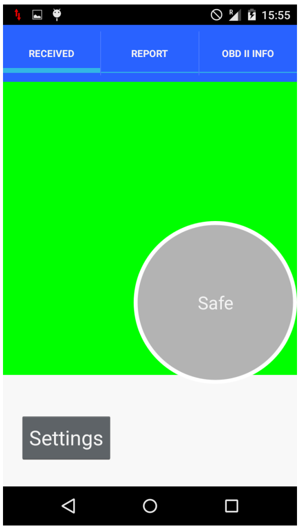

Figure 4.10: Receive Activity

The first tab includes the Receive activity, it is described on 4.10 figure. This Receive activity is simple, it contains a big square that changes its colour depending on the roads conditions, a grey circle that contains textual information, and a setting button. This activity is created to inform the driver about the hazardous events (RHWs) that occurred on the route of the car. With this help the driver can approach the location carefully. The simplicity of the layout helps the user to have fast information in a visual design. The square changes colour depending the distance from the vehicle to the RHW received message. If the device does not receive any notification, the square takes the colour green with the end to indicate that the route is safe. When a RHW is received at considerable distance the square colour will change to yellow. If the distance to the location of the RHW is close the colour of the square will change to red. All the notification of the events display additional information on the grey circle in a text. This information is defined by the received RHW type and the distance to the driver from the event. If there is no RHW information to display the text from the circle will contain the word "Safe".

Finally, the visual element Setting button is presented. It is recommended by Android guidelines to use a menu button, for setting and other option, located in the action bar. The action bar is a component of the typical Android application design [49] . However, it is not used because it is not really comfortable to drive when a button is used for this function.

The Settings button, when it is tapped, launches the UserSettings activity. The layout of this layout is depicted in figure 4.11. The button is located in the bottom left corner of the layout, and the action bar is deleted. The colour of the Settings button is grey to avoid to attract the driver's attention.

The Settings layout is developed to allow the user to save important datas, such as family or friends contacts that will be notified in case of an accident occurred, the emergency contact (EMS) for the eCall and the countdown time in seconds. This layout also is composed by two buttons. The Cancel button closes the Settings activity without saving data. The Save button closes the Settings activity and saves the preferences, displayed on the editable fields, into a database. This process is also explained in section 4.2.1.

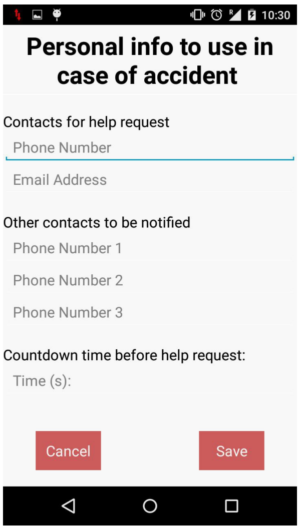

Figure 4.11: Settings Activity

The Report activity depicted in figure 4.12 is also integrated in the tab navigator. This activity is created with the goal to allow the driver to report RHWs to other vehicles via VANET. This layout follows the mockup proposed in section 3.7.1 and described in figure 3.6. The Report Activity layout contains a listview with white background and black text. When one of the items of the listview is tabbed the RHW is reported.

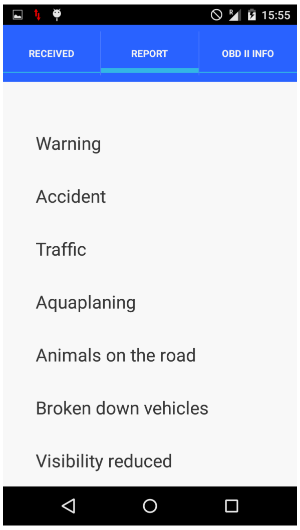

Figure 4.12: Report Activity

The last tab, described on 4.13, integrates the OBD-II Activity. This layout displays the information of the vehicle sensors. The followed design is the one described on the section 3.7.1 and it is presented in the figure 3.8. It is composed by a listview composed by different items that will display the information of vehicle sensor. These items are composed by two texts, one for the title of the name of the sensor and the other for the value of the sensor.

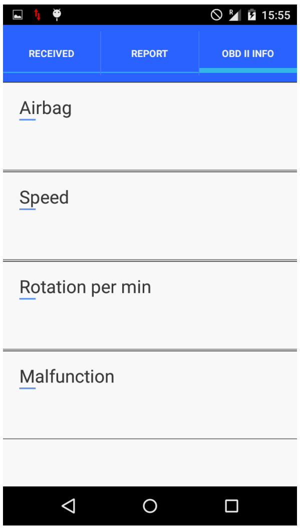

Figure 4.13: OBD-II Activity

The passenger detection layout described in the Figure 4.14 is a resource to wait to the passengers of the vehicle to take a picture with the front camera of the smartphone and detect how many people are in the vehicle. This layout follows the GUI mockup from the section 3.7.1 and described on the figure 3.10. This Layout is only composed by two texts, on the one hand, there is an advertisement explaining to the user that there is a passenger detection running. On the other hand, there is a text that describes how many passengers has the device detected.

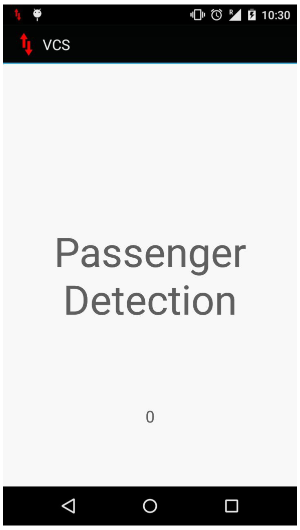

Figure 4.14: Passenger detection Activity

The next layout depicted in Figure 4.15 composed the Countdown layout. This layout pops up when the ADA detects an accident. This activity gives some time to the user to abort the help request. The Countdown activity layout is depicted in figure 4.15. As it is shown, the layout is formed by a countdown timer and a Cancel button. In the case that the user wants to abort the help request and close the activity the Cancel button must be pushed. If the countdown is not interrupted, the algorithm proceeds, as described in section 3.7.1, and the layout changes, by informing the user that help was requested.

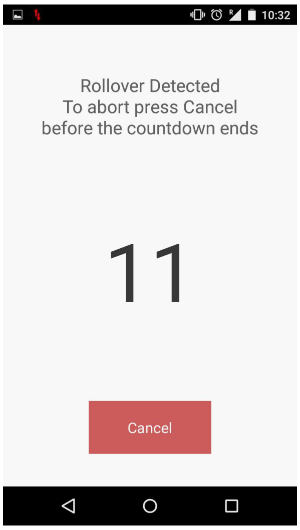

Figure 4.15: Count Down Activity

One of eCall++'s main characteristics is the ADA, inhered from HEADWAY. To execute this algorithm is necessary to access the GPS hardware. In the case that the GPS location system is not enabled, the user is requested to enable it with a dialogue box. This dialogue box is depicted in figure 4.16.

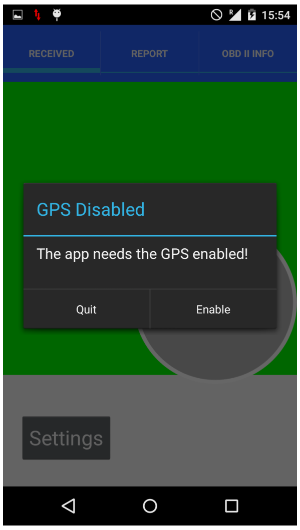

Figure 4.16: GPS enable dialogue box

There are two layouts more to suggest to the user what is happening in two distinct situations. These layouts are Warning layout and Goodbye layouts. Both belong to the MainActivity and they are depicted in figures 4.17 and 4.18 respectively.

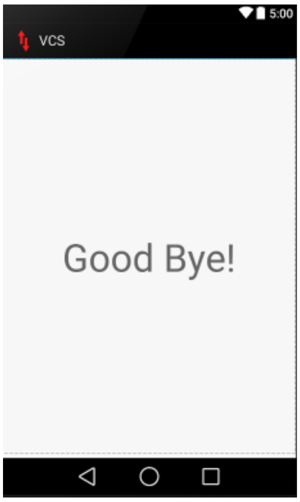

Figure 4.17: Good bye Activity

Figure 4.18: OBD No connect Activity

The "OBD No connect Activity" is intended to notify the user how to proceed, when the user tries to play the eCall++ without connection to the IT2s platform via USB. The Goodbye activity is intended to suggest to the user that eCall++ is closing on purpose.

This chapter has described the GUI design and features of the application, both in a behavioural and structural view. The mechanisms of passenger detection and not continuous video streaming were described in a detailed and a simple way, hiding irrelevant details.

After the application developing, it is appropriate to go about to the testing phase, with the goal to validate the functionalities of eCall++. The next chapter presents the tests performed and its results.

The accident detection algorithm was already validated in HEADWAY project [17] and because of that is not included in this documentation.

CHAPTER 5

Test and validation

After the implementation of eCall++, it is necessary to test the application, with the intention to prove if the goals of the project were obtained and if the system actually works as expected.

This section has the intention of presenting some test performed to the eCall++ in order to prove the passenger detection and the video streaming mechanisms. The results are textually described and aided by pictures and tables with valuable information.

5.1 Passenger detection mechanism test

One of the main characteristic of eCall++ is the passenger detection. This mechanism is able to detect how many people are inside the vehicle. Several tests were performed to assess if the functionality works properly, detecting all kinds of persons in different environments.

As described in section 4.2.1 the passenger detection mechanism uses the Camera of smartphone to detect the number of occupants inside the vehicle.

A special application was developed for this test with the same mechanism. This application has as an output a photography of the inside of the car and a text file with the number of occupants detected in the car.

The chosen scenario is in a *Renault Megane* from the year 2009. It is a current car with 5 seats. The smartphone was located in the front of the car to have a good point of view for this test. Observing the figure 5.1 the main problem of this scenario could be detected: the camera has a huge problem to detect the occupants behind the pilot and copilot seats, the area with light green colour. This test was done with real people.

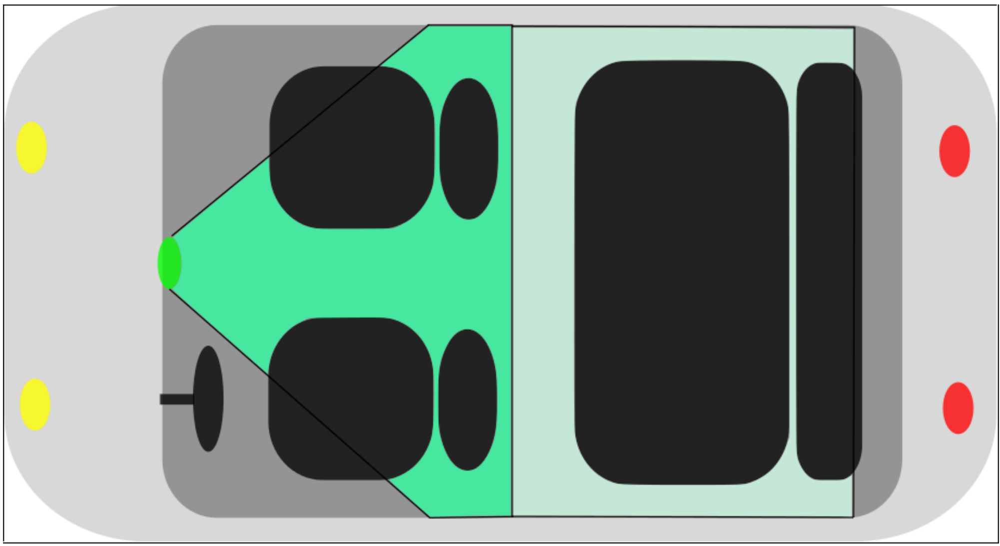

Figure 5.1: Scenario of passenger detection test

There are two different types of test that have been chosen. One of the test was done where the vehicle was exposed to sunlight and other where the vehicle was with cars inside light on during the night in the darkness. In these both scenarios had been performed the same test where the car was occupied from 1 occupant to 5 passengers. In each case there was taken 20 photographs, in total 200 pictures. Following the passenger detection results are presented.

5.1.1 PASSENGER DETECTION RESULTS

In this section, the results of passenger detection inside a car are presented. There are two different main scenarios, one with sunlight and the other is done during the night. In table 5.1 the results of passenger detection test during the sunlight are presented.

	Number of passenger detection with 1 occupants	**Number of passenger detection with 2 occupants**	**Number of passenger detection with 3 occupants**	**Number of passenger detection with 4 occupants**	**Number of passenger detection with 5 occupants**
Photo 1	1	2	2	3	1
Photo 2	1	2	1	2	4
Photo 3	1	2	2	2	5
Photo 4	2	0	1	2	3
Photo 5	1	0	1	2	2
Photo 6	1	1	2	1	3
Photo 7	1	2	2	1	1
Photo 8	1	2	2	2	3
Photo 9	1	2	3	1	3
Photo 10	1	1	3	3	3
Photo 11	1	1	3	2	2
Photo 12	1	1	3	2	2
Photo 13	1	2	3	3	3
Photo 14	1	2	3	2	2
Photo 15	1	2	2	1	2
Photo 16	1	2	3	3	2
Photo 17	1	2	3	2	1
Photo 18	1	1	3	3	3
Photo 19	1	2	3	3	2
Photo 20	1	2	3	4	3

Table 5.1: Passenger detection test during the day light

In table 5.2 the results of passenger detection test during the night with cars inside light on, are presented.

	Number of passenger detection with 1 occupants	**Number of passenger detection with 2 occupants**	**Number of passenger detection with 3 occupants**	**Number of passenger detection with 4 occupants**	**Number of passenger detection with 5 occupants**
Photo 1	1	2	3	2	4
Photo 2	1	1	2	2	4
Photo 3	1	2	2	2	4
Photo 4	1	2	2	3	3
Photo 5	1	2	3	2	3
Photo 6	0	1	3	2	3
Photo 7	0	2	2	2	3
Photo 8	1	2	2	2	3
Photo 9	0	2	2	2	2
Photo 10	1	1	3	2	2
Photo 11	0	2	2	3	3
Photo 12	1	1	2	2	6
Photo 13	0	1	3	3	3
Photo 14	1	2	3	3	3
Photo 15	1	2	3	4	2
Photo 16	1	2	2	4	3
Photo 17	1	2	2	4	3
Photo 18	1	0	1	3	3
Photo 19	1	2	2	2	4
Photo 20	1	1	2	3	2

Table 5.2: Passenger detection test during night light

ASSESSMENT OF THE OBTAINED RESULTS

Observing both tests in some cases the mechanism is not able to detect some faces. This means that in some photographies the face of the passenger is not well defined and the mechanism is not able to detect it. As it is possible to see in the next photograph (Figure 5.2) the person that is supposed to be detected is out of focus because he is not still. This doesn't mean that the mechanism is not focusing when a picture is taken. It means that the user is moving when the photograph is taken and because of that the passenger appears out of focus.

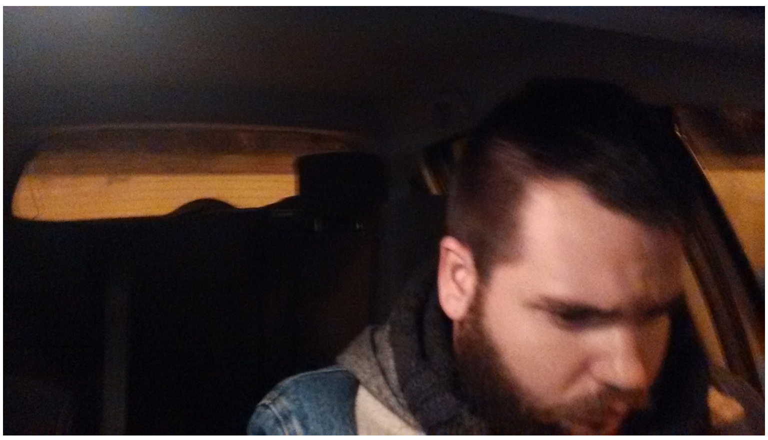

Figure 5.2: Out of focus photograph for passenger detection

The problem of unfocussing happens during light and dark, but it is more common with less light. As it is mentioned before, this mechanism has another problem also; the way of the car distribution does not permit the camera to have a good view to take a picture where all the passengers appear. The two front seats cover up the back passengers as it is possible to see in the figure 5.3.

Figure 5.3: Photograph where the back seat passengers are covered up by the front seats

There are some angels of the face that the mechanism is not able to detect it. One of those examples is seen in the figure 5.4, where the passengers face position is not the best to detect it. The user is not looking at the camera and the mechanism is not able to detect his face.

Figure 5.4: Not good face angle photograph

In some cases, the mechanism is detecting more people than there are. This error is not really usual in the done tests, but in some cases it had happened. In all those photographs where there were detected more people than there was, the pictures were well focused and the peoples faces were in the proper angle. In these cases the passenger detection algorithm was modified to draw a square in the middle of the detected face. In this way it could detect what was the problem that confused the mechanism. One of these tests is showed in the figure 5.5.

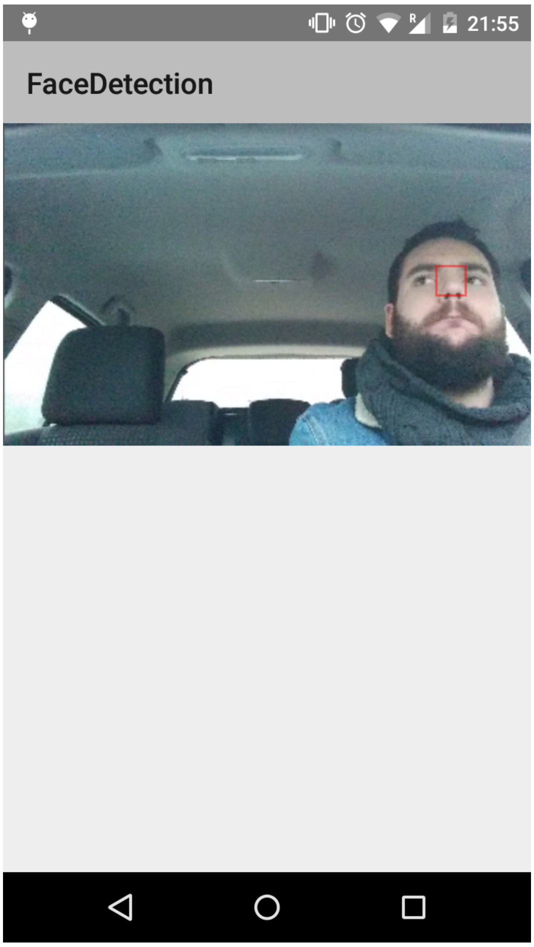

Figure 5.5: Error detection application

As it is possible to see in the figure 5.5, the mechanism does not detect the error and only detects the real face. It is due to the *FaceDetector.Faces* array in *FaceDetector* class, the eye middle point info does not appear on this array.

The mechanism used in this test is not added to the eCall++ because this type of algorithms different estimates and draws wasted a lot of time and became really uncomfortable use for the user. Some times it wastes that much time that the device blocks. This algorithm also needs a powerful processor to process the image. If this mechanism is included to the eCall ++, this will not be able to confront this functionality in all Android devices.

Once these problems have been detected in the mechanism, it has been decided to use the previous results to deal arithmetic averages. This will help to know what average is better for each case. In this way, it will be known what datas are more useful for updating the passenger detection mechanism.

ARITHMETIC AVERAGES FROM RESULTS

After the obtained results, have been decided to calculate some different arithmetic averages to obtain more accurate mechanism for passenger detection inside the car. The arithmetic averages used for this dissertation are really common in statistics. The used average are the following ones:

	Upper value	Lower value	Arithmetic Mean	Median	Mode
During the day light					
1 passenger test	2	1	1.05	1	1
2 passengers test	2	0	1.55	2	2
3 passengers test	3	1	2.4	3	3
4 passengers test	4	1	2.2	2	2
5 passengers test	5	1	2.5	2,5	3
During the night light					
1 passenger test	1	0	0.75	1	1
2 passengers test	2	0	1.6	2	2
3 passengers test	3	1	2.3	2	2
4 passengers test	4	2	2.6	2	2
5 passengers test	6	2	3.15	3	3

Table 5.3: Test datas and arithmetic average comparison

Observing the comparison table 5.3, some datas and averages are not really useful. The lower value, geometric mean, median and the harmonic mean do not help to determinate the passenger detection inside of the car. However, an interpretation between the upper value and the arithmetic mean could help to fix a number of passengers in the vehicle.

The 5.3 is divided in two, on one hand, the day light test datas and on the other hand the night light test datas. They should be analysed in different ways.

The day light test datas, only in the case of "one passenger test" the upper value is wrong. In that case the algorithm should check the arithmetic mean average. If this values round is 1 there is only one person in the vehicle, in the opposite side if it the value 2 there will be two passengers.

On the opposite scenario, the upper value almost in all cases is right. Only in the " five passengers test " is the upper value wrong. The median and any average do not help to differentiate the scenario with 5 passengers.

Finally, the maximum value almost in all cases is accord with the real number of passengers and this averages are only useful in a real scenario where there is a long trip to drive. This eCall++ version is designed for a laboratory environment. This tests are done to scope how should be the future versions of the eCall and to find out the efficiency of the actual application.

5.2 VIDEO STREAMING MECHANISM

The purpose of this dissertation was not to test the communication between AUs and RSU in the VANET. However, even if it is out of scope, how the quality of the video streaming is studied. This test is done in a laboratory environment, so it is not a real case study.

The test has been made between a smartphone and two IT^{2}S platforms. Observing to the figure 5.6, the smartphone is working as an AU. It is connected to an IT^{2}S platform and this one to an another IT^{2}S platform. This video streaming is an unidirectional way. The smartphone sends the video file to the IT^{2}S platform via USB and this one, as OBU device, sends the file divided in packets of 256 bytes to the other IT^{2}S platform via NON-IP protocol. In this connection there is no security implemented.

The first IT^{2}S platform, the OBU, uses the API from *Instituto de Telecomunicações* from *Universidade de Aveiro* called *IT-WiFi* to send the video when it is received from the smartphone. The received video is properly divided in packets and it is send to the other IT^{2}S platform that is working as an RSU. The RSU also uses the *IT-WiFi* to receive the sent video from the OBU.

The communication between this two platforms is centred 5.9 GHz and both platforms are equipped with DSRC 5.9 GHz antenna for this connection. This test also follows the IEEE 802.11g standard as the proposed network in section 3.1. The communication used in this test has not a feed back from the other connected platform if a packet is lost there is no way to recover it.

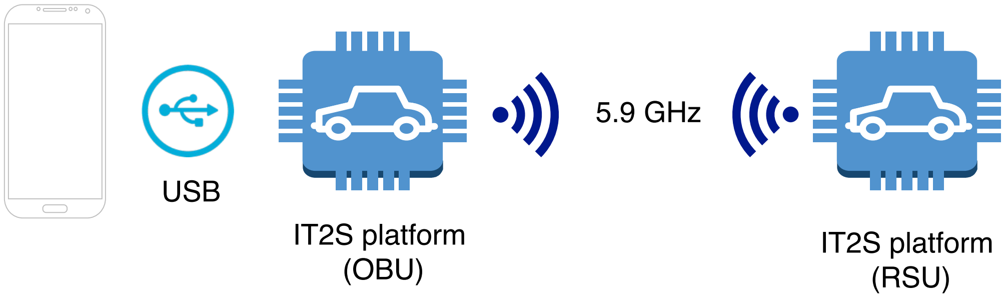

Figure 5.6: Out of focus photograph for passenger detection

The Android application used for this test only sends one live video file to the OBU platform. In this way all the details of the connection could be understand step by step. The OBU also only sends one video file to the RSU because of this.

5.2.1 VIDEO STREAMING QUALITY TEST

The study of quality loss in the wireless communication in VANET environment is out of scope of this dissertation but it was decided to study because it could affect in the future work. In order tot test the missed information of the streamed video the arithmetic median of 35 streamed videos Packet Error Rate (PER) had been done. In the following table 5.4, it is able to read the number of bits and packets sent and received, the PER of each video and the arithmetic median of all PERs. In this test the transmission time that the OBU uses for the video streaming was calculated.

Test number	TX bits	RX bits	TX packets	Lost packets	PER
1	12147614	12139166	5932	5	0.084
2	12147614	12140446	5932	4	0.067
3	12147614	12140702	5932	4	0.067
4	12506331	12500443	6107	3	0.049
5	12506331	12499419	6107	4	0.065
6	12506331	12498139	6107	4	0.065
7	12506331	12498907	6107	4	0.065
8	12506331	12496859	6107	5	0.081
9	12506331	12499419	6107	4	0.065
10	12506331	12498395	6107	4	0.065
11	12506331	12500699	6107	3	0.049
12	12506331	12500443	6107	3	0.049
13	12506331	12499931	6107	4	0.065
14	12506331	12500187	6107	3	0.049
15	12506331	12498395	6107	4	0.065
16	12506331	12497883	6107	5	0.081
17	12506331	12500699	6107	3	0.049
18	12506331	12499931	6107	4	0.065
19	12506331	12496603	6107	5	0.081
20	12506331	12500187	6107	3	0.049
21	12506331	12500699	6107	3	0.049
22	12506331	12498651	6107	4	0.065
23	12506331	12497115	6107	5	0.081
24	12506331	12497627	6107	5	0.081
25	12506331	12498395	6107	4	0.065
26	12506331	12501723	6107	3	0.049
27	12506331	12499419	6107	4	0.065
28	12506331	12500699	6107	3	0.049
29	12506331	12502235	6107	2	0.032
30	12506331	12499931	6107	4	0.065
31	12506331	12500955	6107	3	0.049
32	12506331	12501467	6107	3	0.049
33	12506331	12499419	6107	4	0.065
34	12506331	12501723	6107	3	0.049
35	12506331	12500699	6107	3	0.049
				Total PER:	0.0614

Table 5.4: Results of non-IP video streaming test

Observing to the table 5.4, the total PER of this test is 0.0614%. This value is low but it would be higher in a motorway environment where the communication is in a moving environment, with numerous obstacles and more streaming sources. It is possible to deduce from these results that there is no case where the totally of the sent video is received in the other end. The PER value could be lower if a protocol that ensures the recovery of lost packets is used for this communication. This test also verifies that the streamed videos have an acceptable PER after emission and that they are able to play in the RSU after the streaming.

Finally, in this test the transmission time that the OBU needs to send to the RSU in a file of 12,506331 Megabits was calculated. This value was of 73500 milliseconds. This is a loss of time really, 1 minute 13 seconds and 500 milliseconds. There is spent 12.035 seconds for buffering and transmitting each packet video. This is a problem that could block the communication and it should be improve in following versions of the *IT-WiFi* API. This time is also related to the size of the used packets, if this size is bigger the sending time will be decrease.

CHAPTER 6

CONCLUSION

The main purpose of the eCall++ was to create an AU for VANET and to figure out the capabilities of this type of networks. In this project the potential abilities of VANET have been demonstrated.The eCall++ had demonstrated that an application with a passenger detection and video streaming mechanism is able. This project also has shown how an AU based on a smartphone is able. These mechanisms have been able due to the rich hardware and software resources of the smartphones.

The eCall ++ also has been developed an application based on the eCall system. These permits to the driver receive emergency aid in case of accident.

The vehicular communication system proposed in this dissertation uses Facilities layer from the ETSI protocol stack. Because of this, DENM and CAM messages are sent between the OBUs. The objective of exchanging information between vehicles from the highway has been addressed.

With these all implemented features the smartphone can be valued ITS system that archives the goals of becoming the driving more safety, comfortable and efficient.

eCall++ had develop the functionalities as expected and it was implemented successfully.

6.1 LIMITATION AND CHALLENGES OF ECALL++ FOR VEHICULAR NETWORKS

The mobile application eCall++ in vehicular networks has some limitations that must be solved over time. They are following numbered:

- **Passenger detection in blind spots:** the eCall++ presents some blind spots when a photograph is taken for passenger detection. This should be solved taking more photographs and detecting on which seats are the passengers inside the car. This counts where is each occupant and it does not repeat the passengers during the count.
- **Good quality photographs during the night light:** the darkness inside the car makes it impossible to take good quality photographs with passenger detection mechanism. Without a good quality image the face detection mechanism does not work that well. With a night vision camera this problem could be solved.

- **Execution of two cameras in the smartphone at the same time:** nowadays it is not possible to create an application which is able to use the front and back cameras in an smartphone. It doesn't matter if it is an Android or iOS smartphone, this problem exists in both OS. So the cameras should be switched between them to develop this type of application, but it becomes the application really slow.
- **Baby sit detection:** with the passenger detection mechanism proposed in the eCall++ is difficult to detect a baby. Only if the baby is looking to the camera, it could be detected. However, the majority of baby seat systems are designed to position the baby looking to the back side of the car. Because of this it was impossible to detect a baby with this system.

Some of the limitations and challenges described in this section could be solved with a new version of the eCall++. There are other limitations such as the two cameras execution that couldn't be solved in a new version.It is because it depends on the smartphone manufacturers.

The goals that could be developed in a new version of the eCall++ are reflected in the following 6.2 section.

6.2 FUTURE WORK

As in previous section were directly or indirectly mentioned, several aspects should be improved in the future, to became the eCall++ a skilled commercial application and leave the proof of concept stage. The main improvements are described following:

- **Continuous Live Streaming mechanism:** with a continuous live streaming mechanism in case of accident the mobile phone could send a continuous information to the first aid centre, with out slicing the video.
- **Improve the video file:** a better video streaming is possible if the video file moov atom header is at the beginning of the .mp4 file. The problem is that there is no way to make this type of file using Android in this moment. Neither *MediaRecorder* class or other class of Android are able to create a video with the moov atom header at the beginning of the file. The point of this issue if that if the header is located at the start of the file the receiving device is able to start playing the video without downloading all the file. Another solution could to address this goal could be to modify the video on the sending OBU, but it will be faster if it was created with the appropriate format from the source of the video.
- **Improve the video streaming communication:** improving this communication the video streaming could be faster than in this moment it is. An error correction protocol could be used to address this goal.
- **Add voice warnings for RHW:** with this functionality the driver could drive without loosing the view on the road.
- **Receiving video streaming:** if the driver could see the real state of the road on a video before starting to drive, a better route could be chosen by the driver.
- **Real case should performance:** a real test could find out more limitations and challenges of the eCall ++.

- **Develop a more minimalist Layout:** a better design could do the driving more comfortable without loosing the focus on the road.
- **Enable and disable application services:** developing more settings and giving the options to the user to enable and disable services.
- **Driving automatic detection:** detects when the user is driving and blocks the application. The application is only going to show the warnings of the road. If the user wants to send any information about an accident he must stop driving. This new service will make the application more safe.
- **Implement an automatic detection of the light state:** if the application is running during day or night or if is inside a tunnel it should work in a different way to show the layout in a better way or detect the passengers with a different mechanism.
- **Update a passenger detection mechanism:** uses the eye middle value and the arithmetic average to perform a passenger detection mechanism that could be executed during the driving in a real case. This mechanism was not developed in the eCall ++ due to the low yield that it involves to nowadays devices.

Bibliography

[1] E. Commision. (Mar. 2015). Road safety in the European Union, Trends, statistics and main challenges, [Online]. Available: http://ec.europa.eu/transport/road_safety/pdf/vademecum_2015.pdf.

[2] E. T. S. Institute. (2016). ETSI, [Online]. Available: http://www.etsi.org.

[3] I. of Electrical and E. Engineers, Eds. (2016). IEEE, [Online]. Available: https://www.ieee.org/index.htmlhttps://www.ieee.org/index.html.

[4] S. Antipolis, "European Parliament makes eCall mandatory from 2018", May 2015. [Online]. Available: http://www.etsi.org/news-events/news/960-2015-05-european-parliament-makes-ecall-mandatory-from-2018.

[5] E. Commission, "ECall in all new cars from April 2018", Apr. 2015. [Online]. Available: https://ec.europa.eu/digital-agenda/en/news/ecall-all-new-cars-april-2018.

[6] *ECall*, iCar Support, Jan. 2016. [Online]. Available: http://www.icarsupport.eu/ecall/.

[7] E. C. for Standardization, *Intelligent transport systems - esafety - ecall minimum set of data, standard: cen - en 15722*, Mar. 2015.

[8] eCall Driving Group, *Recommendations of the DG eCall for the introduction of the pan-European eCall,* Technical report, 2016.

[9] R. J. Weiland, "Intelligent Transportation Systems".

[10] L. Figueiredo, "Sistemas Inteligentes de Transporte", PhD thesis, Faculdade de Engenharia da Universidade do Porto, Feb. 2005. [Online]. Available: http://ave.dee.isep.ipp.pt/~gris/_private/Teses/Thesis_PhD_Lino2005.pdf.

[11] D. of Infrastucture and R. Development, *Benefits of Intelligent Transport Systems*, Australian Government, Nov. 2015. [Online]. Available: https://infrastructure.gov.au/transport/its/benefits.aspx.

[12] Y. Z. H. Moustafa, "Vehicular Networks, Techniques, Standards and Applications", PhD thesis, Auerbach Publications, Nov. 2008.

[13] A. M. C. Campolo, "Multichannel Communications in Vehicular Ad Hoc Networks: A Survey.A Survey", IEEE Communications Magazine, 2013.

[14] E. T. S. I. (ETSI), "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service", Mar. 2011.

[15] "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service", Sep. 2013.

[16] IDC, “Smartphone Vendor Market Share, 2015 Q2”, Aug. 2015. [Online]. Available: http://www.idc.com/prodserv/smartphone-market-share.jsp.

[17] B. Inovacaõ, “HEADWAY-Connecting vehicles and highways”, Aug. 2015. [Online]. Available: http://www.%20brisainovacao.pt/en/innovation/projects/headway.

[18] Android, *Android developer*, 2016. [Online]. Available: http://developer.android.com/.

[19] R. Meier, “Professional AndroidTM 4 Application Development”, 2012.

[20] Automatic, Feb. 2016. [Online]. Available: https://www.automatic.com/home/.

[21] W. B. N. A. of Insurance Commissioners (NAIC), Feb. 2016. [Online]. Available: https://itunes.apple.com/us/app/wreckcheck/id543290503?mt=8.

[22] N. A. of Insurance Commissioners, Feb. 2016. [Online]. Available: http://www.naic.org.

[23] A. Auto, Feb. 2016. [Online]. Available: https://play.google.com/store/apps/details?id=com.google.android.projection.gearhead.

[24] P. C. Safe, Feb. 2016. [Online]. Available: http://www.peugeotcarsafe.com.ar/download_app.jsp.

[25] WAZE, Feb. 2016. [Online]. Available: https://www.waze.com.

[26] P. C. Connect, Feb. 2016. [Online]. Available: http://www.porsche.com/international/carconnect/.

[27] Jules White, Chris Thompson, Hamilton Turner, Brian Dougherty, and Douglas C. Schmidt, “WreckWatch: Automatic Traffic Accident Detection and Notification with Smartphones”, 2007.

[28] SOSmart, Feb. 2016. [Online]. Available: http://www.sosmartapp.com.

[29] iOnRoad Aumented driving, Feb. 2016. [Online]. Available: https://play.google.com/store/apps/details?id=com.picitup.iOnRoad.pro.

[30] C. call, Feb. 2016. [Online]. Available: http://www.collisioncall.com.

[31] C. pro (DashCam & OBD), Feb. 2016. [Online]. Available: https://play.google.com/store/apps/details?id=com.pokevian.prime&hl=es.

[32] SocialDrive, Feb. 2016. [Online]. Available: http://www.socialdrive.es.

[33] SaveDrives, Feb. 2016. [Online]. Available: http://www.savedrives.com.

[34] C. logística e segurança, Feb. 2016. [Online]. Available: http://www.carsafe.com.br/index.html.

[35] T. M. Corporation, Sep. 2015. [Online]. Available: http://newsroom.toyota.co.jp/en/detail/9676551/.

[36] Xerox, *Xerox Rolls Out Vehicle Passenger Detection System with Highway Speed Accuracy for HOV/HOT Lane Enforcement*, Feb. 2016. [Online]. Available: http://news.xerox.com/news/Xerox-Vehicle-Passenger-Detection-System.

[37] “Delphi Passive Occupant Detection System-B”, 2001. [Online]. Available: http://www.delphi.com/docs/default-source/old-delphi-files/6653a4f0-6c6d-4295-8589-3abf4659ee1d-pdf.

[38] Nidec, *ODS: Occupant Detection System*, 2016. [Online]. Available: http://www.elesys-na.com/occupant-detection#3.

[39] J. Krumm and G. Kirk, “Video Occupant Detection for Airbag Deployment”, Oct. 1998. [Online]. Available: http://research.microsoft.com/pubs/68236/airbag.pdf.

[40] S. V. Wilson S. Leoputra and T. Tan, “Passenger Monitoring in Moving Bus Video”, Dec. 2008.

[41] T. Izumi, “Development of Occupant Detection System Using Far-Infrared Ray (FIR) Camera”, Oct. 2009. [Online]. Available: http://global-sei.com/technology/tr/bn69/pdf/69-10.pdf.

[42] M. Klomark, “Occupant Detection using Computer Vision - Master’s Thesis in Computer Vision at Linköping University”, May 2000. [Online]. Available: http://www2.cvl.isy.liu.se/ScOut/Masters/Papers/Ex3026.pdf.

[43] *Our project*, ICSI, Feb. 2016. [Online]. Available: http://www.ict-icsi.eu/description.html#.Vp_MGzZgbIM.

[44] ICSI, “Intelligent cooperative sensing for improved traffic efficiency - system architecture”, Apr. 2013. [Online]. Available: www.ict-icsi.eu.

[45] Matos J., Oliveira A., Meireles T., Ferreira N., Mar P., Fonseca J., Carona D., Serrador A., Lopes J., “Emergent vehicular communications: applications, standards and implementation”, *Proc URSI Seminar of the Portuguese Committee*, vol. 1, Sep. 2010.

[46] Joaquim Ferreira, Arnaldo Oliveira, Joaõ Almeida, and Cristóvaõ Cruz, “Fail silent road side unit for vehicular communications”, *Proceedings of Workshop ASCoMS (Architecting Safety in Collaborative Mobile Systems) of the 32nd International Conference on Computer Safety, Reliability and Security*, 2013.

[47] M. J. N. Oliveira A. S. R., “IT2S Board Description”, Mar. 2013.

[48] Vítor Emanuel Ornelas Gomes, “Smartphone based accident detection and eCall implementation”, 2013.

[49] A. design web page, Feb. 2016. [Online]. Available: http://developer.android.com/intl/es/design/index.html.

[50] UML®, Jan. 2016. [Online]. Available: http://www.uml.org.

[51] Android, *FaceDetector*, Jan. 2016. [Online]. Available: http://developer.android.com/intl/es/reference/android/media/FaceDetector.html.

[52] *FaceDetector.face*, Android, Jan. 2016. [Online]. Available: http://developer.android.com/intl/es/reference/android/media/FaceDetector.Face.html.

[53] *Controlling the Camera*, Android, Jan. 2016. [Online]. Available: http://developer.android.com/intl/es/training/camera/cameradirect.html.

[54] *Camera*, Android, Jan. 2016. [Online]. Available: http://developer.android.com/reference/android/webkit/WebView.htm.

[55] *Camera2*, Android, Jan. 2016. [Online]. Available: http://developer.android.com/intl/ru/reference/android/hardware/camera2/package-summary.html.